(19) BUNDESREPUBLIK DEUTSCHLAND

DEUTSCHES PATENTAMT

(12) Patentschrift

(10) DE 195 48 387 C 1

(51) Int. Cl.[6]:
H 04 L 9/00
G 06 F 12/14
G 06 F 13/42

(21) Aktenzeichen: 195 48 387.1-31
(22) Anmeldetag: 22. 12. 95
(43) Offenlegungstag: —
(45) Veröffentlichungstag der Patenterteilung: 30. 1. 97

DE 195 48 387 C1

Innerhalb von 3 Monaten nach Veröffentlichung der Erteilung kann Einspruch erhoben werden

(73) Patentinhaber:
Siemens AG, 80333 München, DE

(72) Erfinder:
Pfaff, Oliver, Dr., 10827 Berlin, DE

(56) Für die Beurteilung der Patentfähigkeit in Betracht gezogene Druckschriften:

EP 01 68 667 A2
JP 05-1 51 044 A2
US-Z.: BELLOVIN, S. et al.: Network Firewalls, in: IEEE Communications Magazin, Sept. 1994, S. 50-57;
US-Z.: GARFINKEL, D. et al.: HP SharedX: A Tool for Real-Time Collaboration, in: Hewlett-Packard Journal, April 1994, S. 23-36;

(54) Verfahren zur kryptographischen Sicherung der rechnergestützten digitalen Kommunikation zwischen einem Programm und mindestens einer Benutzereinheit

(57) Da bei vielen Client-Server-Fenster-Systemen die Systeme nur in Objektcode und nicht in Quellcode vorliegen, ist eine Sicherheitserweiterung nur schwer bzw. gar nicht möglich.
Bei dem erfindungsgemäßen Verfahren werden die schon in Transportprotokoliformat (TP) codierten Anforderungen (A) bzw. Nachrichten (B) noch einmal in der Transportprotokollschicht (TP) decodiert, und dann in einer Sicherheitsschicht (SL) beliebigen kryptographischen Verfahren unterzogen. Danach werden sie wieder in der Transportprotokollschicht (TP) codiert und einem Programm (P) oder mindestens einer Benutzereinheit (XS) übertragen.
Damit ist eine Sicherheitserweiterung, beispielsweise hinsichtlich der Verschlüsselungsdaten, der Autentikation oder auch der Zugriffskontrolle, erreicht.

DE 195 48 387 C1

Beschreibung

Die Erfindung betrifft ein Verfahren zur kryptographischen Sicherung der Kommunikation in sogenannten Client-Server-Fenster-Systemen mit einer offenen Netzschnittstelle. Ein Beispiel solcher Client-Server-Fenster-Systeme ist beschrieben in [1].

Durch eine zusätzliche Verwendung einer sogenannten Application Sharing Komponente, die dazu verwendet wird, Anforderungen von Benutzereinheiten an das Programm zu multiplexen und Nachrichten von dem Programm, zum Beispiel Ereignismeldungen, Antworten oder Fehlermeldungen, für die Benutzereinheiten zu demultiplexen, wird die gemeinsame Nutzung einer Standard-Ein-Benutzer-Anwendung (Standard Single User Application) zwischen verschiedenen heterogenen Umgebungen, die sich an unterschiedlichen Orten befinden können, erreicht.

Diese gemeinsame Bearbeitung eines Programms wird als Application Sharing bezeichnet.

Um jedoch auch eine verläßliche Kommunikation vertraulicher Daten zu erreichen, muß das in [1] beschriebene Client-Server-Fenster-System um kryptographische Merkmale erweitert werden.

Dies ist von besonderer Bedeutung bei unternehmensübergreifender Kommunikation. Dies bedeutet bei einer Kommunikation zwischen Rechnern, bei denen sich ein Rechner im prinzipiell abgesicherten vertrauenswürdigen sogenannten Corporate Network eines Unternehmens befindet, und anderen Rechnern, die sich in einer gemeinsamen synchronen verteilten Arbeitsumgebung mehrerer vernetzter Rechner, in einem sogenannten Computer System Cooperated Work-System (CSCW-Systemen), welches Application Sharing realisiert, befindet, nur über einen unsicheren Kanal erreicht werden kann, wodurch eine sichere Kommunikation nicht mehr gewährleistet ist.

Solche CSCW-Systeme basieren auf der Möglichkeit, eine Standard-Ein-Benutzer-Anwendung (Standard Single User Application) gemeinsam mit anderen Benutzern zu einem Zeitpunkt zu bearbeiten.

Die zwischen den Rechnern ausgetauschte Information kann von besonderer Bedeutung sein, beispielsweise kann es sich um vertrauliche Geschäftsinformation handeln, um Designspezifikationen, Finanztransaktionen oder um medizinische Daten, die über den unsicheren Kanal ausgetauscht werden.

Aus diesem Grund ist es notwendig auch für diese Transaktionen von Anwendungsdaten eine gewisse Sicherheit zu gewährleisten.

Bei vielen kommerziellen Systemen, die auf dem in [1] beschriebenen Client-Server-Fenster-System beruhen, ist eine direkte Integration kryptographischer Merkmale nicht möglich.

Der Erfindung liegt somit das Problem zugrunde, ein Verfahren zur kryptographischen Sicherung der rechnergestützten digitalen Kommunikation zwischen einem Programm und mindestens einer Benutzereinheit, anzugeben.

Das Problem wird durch das Verfahren gemäß Patentanspruch 1, das Verfahren gemäß Patentanspruch 4, das Verfahren gemäß Patentanspruch 7 sowie das Verfahren gemäß Patentanspruch 8 gelöst.

Bei dem Verfahren gemäß Patentanspruch 1 wird von dem Programm eine Nachricht gebildet, die für ein Transportprotokoll codiert wird. Direkt nach der Codierung wird unter Verwendung des Transportprotokolls die codierte Nachricht wieder decodiert und die decodierte Nachricht einem kryptographischen Verfahren unterzogen. Danach wird die Anforderung wiederum mit dem Transportprotokoll codiert und an mindestens eine Benutzereinheit übertragen. Hierbei können sich das Programm und die Benutzereinheit auf einem oder auch auf verschiedenen Rechnern befinden.

Bei dem Verfahren gemäß Patentanspruch 4 werden prinzipiell dieselben Schritte ausgeführt, mit dem Unterschied, daß diesmal eine Anforderung in einer Benutzereinheit gebildet wird und dort auch die im vorigen beschriebenen weiteren Schritte durchgeführt werden. Zum Schluß des Verfahrens wird in diesem Fall die codierte kryptographische verarbeitete Anforderung zu dem Programm übertragen.

Bei dem Verfahren gemäß Patentanspruch 7 wird von der Situation ausgegangen, daß auf der Seite des Programms eine Erweiterung des Client-Server-Fenster-Systems durch Sicherheitsmechanismen unterschiedlichster Art, die im weiteren beschrieben werden, möglich ist. Für diesen Fall werden die im vorigen beschriebenen Verfahrensschritte ausgehend von der Bildung einer Nachricht in dem Programm nur nach Empfang der kryptographisch in der eingefügten Sicherheitsschicht auf der Seite des Programms bearbeiteten Anforderungen in einer der Benutzereinheiten durchgeführt. Dort werden also die inversen kryptographischen Verfahren zur Bearbeitung der Nachrichten durchgeführt, wobei dieser Verfahrensschritt durch eine zuvor erfolgte Decodierung mit dem Transportprotokoll und eine nachfolgende Codierung mit dem Transportprotokoll charakterisiert ist.

Das Verfahren gemäß Patentanspruch 8 weist prinzipiell die gleichen Verfahrensschritte auf wie das Verfahren gemäß Patentanspruch 7 mit dem Unterschied, daß hierbei eine Anforderung von einer Benutzereinheit gebildet wird und an das Programm übertragen wird. Die Stellen, an denen die einzelnen Verfahrensschritte ablaufen, und die Stellen, an denen die Verfahrensschritte des Patentanspruchs 7 ausgeführt werden, sind in diesem Fall einfach vertauscht.

Vorteilhafte Weiterbildungen der erfindungsgemäßen Verfahren ergeben sich aus den abhängigen Ansprüchen.

Es ist vorteilhaft, als kryptographische Bearbeitung zumindest eine Verschlüsselung der Anforderung vorzusehen. Damit wird die Vertraulichkeit der ausgetauschten Daten gewährleistet.

Weiterhin ist es vorteilhaft, als kryptographische Bearbeitung der Anforderung Integritäts- und Authentikationsmechanismen vorzusehen, wodurch dann jeweils gewährleistet ist, daß die empfangene Nachricht tatsächlich von dem Absender stammt, der auch in der Nachricht als Absender angegeben ist.

In einer Weiterbildung der erfindungsgemäßen Verfahren ist es außerdem vorteilhaft, als kryptographische Verfahren Zugriffskontrollmechanismen zu realisieren, um somit sicherzustellen, daß wirklich nur diejenigen Anforderungen auch durchgeführt werden, die auch die Berechtigung zur Durchführung aufweisen.

Ferner ist es vorteilhaft vor Beginn des Verfahrens in einer Initialisierungsphase beispielsweise die kryptographischen Schlüssel, die zur Realisierung der einzelnen kryptographischen Verfahren eingesetzt werden, auszutauschen zwischen dem Programm und der mindestens einen Benutzereinheit.

Eine vorteilhafte Verwendung der Verfahren findet sich beim Datenaustausch zwischen Kommunikationspartnern, die über die Grenzen eine Coporate Net-

works, welches durch kryptographische Verfahren in sich gesichert ist, über einen unsicheren Kanal in einem sogenannten Firewall.

Durch eine solche Verwendung ist es nicht mehr wie bisher nötig, bei einer vorgesehenen Kommunikation über Coporate Network-Grenzen hinweg den für die Kommunikation zu verwendenden Rechner von dem gesamten Netz des Coporate Networks zu entkoppeln, um somit nicht das gesamte Coporate Network zu gefährden bei möglichen Angriffen über den unsicheren Kommunikationskanal.

In den Figuren sind einige Ausführungsbeispiele dargestellt, die im folgenden näher erläutert werden.

Es zeigen

Fig. 1 das allgemeine Prinzip eines Client-Server-Fenster-Systems;

Fig. 2 das allgemeine Prinzip eines Client-Server-Fenster-Systems in einer "Mehrbenutzer-Umgebung";

Fig. 3 eine Anordnung, die die Mehrbenutzer-Umgebung detaillierter beschreibt;

Fig. 4 ein prinzipielles Blockschaltbild, in dem das Einfügen einer Sicherheitsschicht zwischen Client-Surfer-Fenstersystem und dem Transportprotokoll beschrieben ist;

Fig. 5 eine Anordnung, in der prinzipiell dargestellt ist, wie das erfindungsgemäße Verfahren in einem Firewall zur Kommunikationssicherung über Coporate Network-Grenzen hinweg verwendet werden kann;

Fig. 6 ein Ablaufdiagramm, in dem die Verfahrensschritte des Verfahrens gemäß Patentanspruch 1 dargestellt sind;

Fig. 7 ein Ablaufdiagramm, in dem die Schritte des Verfahrens gemäß Patentanspruch 2 dargestellt sind;

Fig. 8 ein Blockdiagramm, in dem die einzelnen Möglichkeiten zur Realisierung der sicherheitsspezifischen Bearbeitung der Anforderung bzw. der inversen sicherheitsspezifischen Bearbeitung der Anforderung beschrieben ist.

Fig. 9 ein Ablaufdiagramm, in dem die einzelnen Verfahrensschritte des Verfahrens gemäß Patentanspruch 4 dargestellt sind;

Fig. 10 ein Ablaufdiagramm, in dem die Schritte des Verfahrens gemäß Patentanspruch 5 dargestellt sind;

Fig. 11 ein Ablaufdiagramm, in dem die Schritte des Verfahrens gemäß Patentanspruch 7 dargestellt sind;

Fig. 12 ein Ablaufdiagramm, in dem die Schritte des Verfahrens gemäß Patentanspruch 8 dargestellt sind;

Fig. 13 ein Blockschaltbild, in dem die einzelnen zur Durchführung des Verfahrens gemäß Patentanspruch 1 benötigten Komponenten und der Nachrichtenaustausch beschrieben ist;

Fig. 14 ein Blockschaltbild, in dem die einzelnen zur Durchführung des Verfahrens gemäß Patentanspruch 4 benötigten Komponenten und der Nachrichtenaustausch beschrieben ist;

Fig. 15 ein Blockschaltbild, in dem die einzelnen zur Durchführung des Verfahrens gemäß Patentanspruch 2 benötigten Komponenten und der Nachrichtenaustausch beschrieben ist;

Fig. 16 ein Blockschaltbild, in dem die einzelnen zur Durchführung des Verfahrens gemäß Patentanspruch 5 benötigten Komponenten und der Nachrichtenaustausch beschrieben ist;

Fig. 17 ein Blockschaltbild, in dem die einzelnen zur Durchführung des Verfahrens gemäß Patentanspruch 7 benötigten Komponenten und der Nachrichtenaustausch beschrieben ist;

Fig. 18 ein Blockschaltbild, in dem die einzelnen zur Durchführung des Verfahrens gemäß Patentanspruch 8 benötigten Komponenten und der Nachrichtenaustausch beschrieben ist.

Anhand der **Fig. 1** bis 18 wird die Erfindung weiter erläutert.

In **Fig. 1** ist eine Benutzerumgebung dargestellt, die beispielsweise bei einem Client-Server-Fenster-System, welches in [1] beschrieben ist, auftritt.

Diese Anordnung weist mindestens folgende Komponenten auf:

- eine Benutzereinheit XS, im weiteren als auch Server XS bezeichnet, die wiederum folgende Komponenten aufweist:
- mindestens eine Treibereinheit DD, die eine Kopplung zwischen weiteren Peripheriekomponenten mit einem im weiteren beschriebenen Klienten XC ermöglicht,
- eine Bildschirmeinheit BS,
- eine Tastatur TA,
- eine Maus MA,
- den Klienten XC, der mindestens folgende Komponenten aufweist:
- eine Menge von Bibliotheksroutinen XL sowie
- eine Anwendung ANW.

Die Bildschirmeinheit BS, die Tastatur TA, die Maus MA sowie eventuell außerdem vorhandene weitere Peripherieeinheiten bilden die im vorigen beschriebenen Peripheriekomponenten, die über die entsprechenden Treibereinheiten DD mit dem Klienten XC gekoppelt sind.

Die Menge der Bibliotheksroutinen XL des Klienten XC bildet die Schnittstelle zwischen der Anwendung ANW, beispielsweise einem Textverarbeitungsprogramm oder auch einem Tabellenkalkulationsprogramm oder allen anderen bekannten Anwendungen ANW, und der Benutzereinheit XS.

Zusammen bilden die Bibliotheksroutinen XL sowie die Anwendung ANW ein Programm P.

Auch wenn in diesem Ausführungsbeispiel nur jeweils eine Anwendung ANW bzw. ein Programm P beschrieben wird, können selbstverständlich mehrere Anwendungen ANW und damit mehrere Klienten XC auf einer, diese Anwendung ANW ausführenden Rechnereinheit zur Verfügung gestellt werden.

Diese in **Fig. 1** dargestellte Anordnung ist also nur ein sehr einfaches, prinzipielles Beispiel für den Ablauf der Kommunikation eines Klienten XC mit dem Server XS, wie sie unter dem bekannten, in [1] beschriebenen Client-Server-Fenster-System durchgeführt wird.

Von dem Server XS wird eine Anforderung A an den Klienten XC gesendet. Dadurch werden in dem Klienten XC Aktionen, beispielsweise in der Anwendung ANW, angestoßen.

Die Anforderung A kann zum Beispiel eine Eingabe auf der Tastatur TA sein, die durch die Treibereinheiten DD in die Anforderung A "übersetzt" und an den Klienten XC gesendet wird.

Die Anwendung ANW, beispielsweise ein Textbearbeitungsprogramm oder ein Kalkulationsprogramm, ein Zeichenprogramm und ähnliche Programme, kann nun die Eingabe akzeptieren und beispielsweise als einen neuen Buchstaben in der Textdatei aufnehmen.

Damit diese Änderung in der Textdatei auch auf dem Bildschirm BS dargestellt werden kann, wird in einer Antwort B in diesem Fall beispielsweise eine Darstellungsnachricht an die Bildschirmeinheit BS gesendet,

mit der eine Änderung in der Bildschirmdarstellung angefordert wird.

Ein Nachteil vieler kommerzieller Systeme, die nach diesem Prinzip arbeiten, liegt vor allem darin, daß eine direkte Integration benötigter Sicherheitsmechanismen in das Client-Server-Fenster-System oftmals nicht möglich ist.

Hierzu wäre nämlich ein direkter Eingriff in die Schnittstelle zwischen den Bibliotheksroutinen XL und den Transportprotokollen nötig. Diese sind eben oftmals dem Benutzer nicht zugänglich.

In **Fig.** 6 ist ein Ablaufdiagramm mit einzelnen Verfahrensschritten des erfindungsgemäßen Verfahrens gemäß Patentanspruch 1 dargestellt. Die zur Durchführung dieses Verfahrens nötige Anordnung ist in **Fig.** 13 beschrieben.

Von dem Programm P wird in einem ersten Schritt 601 die Nachricht B gebildet.

In einer Transportprotokollschicht TP wird aus der Nachricht B eine neue Nachricht gebildet, in dem die Nachricht B in das Transportprotokollformat "eingebettet", also codiert wird 602, CB.

Eine Übersicht über verschiedene Transportprotokolle ist in [2] zu finden. Die erfindungsgemäßen Verfahren sind unabhängig von dem speziell jeweils verwendeten Transportprotokoll.

Entweder auf derselben Rechnereinheit, auf der das Programm P läuft, oder auf einer gesondert vorgesehenen ersten Sicherungscomputereinheit SC1, die über einen sicheren Kanal mit dem Rechner gekoppelt ist, wird in der dort vorgesehenen Transportprotokollschicht TP die codierte Nachricht CB decodiert 603, DB.

Die decodierte Nachricht DB wird nun einer Sicherheitsschicht SL zugeführt, in der sie unterschiedlichen, beliebig vorgegebenen kryptographischen Verfahren unterzogen wird 604.

Eine durch die kryptographische Bearbeitung gebildete kryptographisch bearbeitete Nachricht VB wird nun wiederum in der Transportprotokollschicht TP codiert 605, wodurch eine codierte kryptographisch verarbeitete Nachrichtung CVB gebildet wird.

Die codierte kryptographisch verarbeitete Nachricht CVB wird in einem letzten Schritt 606 an die Benutzereinheit XS, also an den Server übertragen.

Der prinzipiell umgekehrt gelagerte Fall für die Anforderung A aus **Fig.** 1 ist in **Fig.** 9 in Form eines Ablaufdiagramms und in **Fig.** 14 in Form eines Blockdiagramms für die Anordnung, die zur Durchführung des Verfahrens benötigt wird, dargestellt.

In diesem Fall wird die Anforderung A von der Benutzereinheit XS gebildet 901.

Die Anforderung A wird der Transportprotokollschicht TP zugeführt und dort in das jeweils verwendete Transportprotokollformat eingebettet 902. Eine hieraus resultierende codierte Anforderung CA wird nunmehr entweder in der Benutzereinheit XS selbst oder in einer gesondert vorgesehenen zweiten Sicherungscomputereinheit SC2, die über einen sicheren Kanal mit der Benutzereinheit XS gekoppelt ist, in der Transportprotokollschicht TP "ausgepackt", also decodiert 903, wodurch eine decodierte Anforderung DA gebildet wird.

In der Sicherheitsschicht SL wird die ihr zugeführte decodierte Anforderung DA nunmehr den vorgesehenen kryptographischen Verfahren, die im weiteren beschrieben werden, unterzogen 904. Daraus resultiert eine kryptographisch bearbeitete Anforderung VA.

Die kryptographisch bearbeitete Anforderung VA wird wiederum der Transportprotokollschicht TP zugeführt 905 und dort codiert, wodurch eine codierte kryptographisch bearbeitete Anforderung CVA gebildet wird. Die codierte kryptographisch bearbeitete Anforderung CVA wird in einem letzten Schritt 906 an das Programm P, also an den Klienten XC übertragen.

Eine Weiterbildung des Verfahrens gemäß Patentanspruch 1 ist in **Fig.** 7 in Form eines Ablaufdiagramms, sowie die zur Durchführung dieses Verfahrens notwendige Anordnung in **Fig.** 16 dargestellt.

Nachdem die in **Fig.** 6 dargestellten Verfahrensschritte zur letztendlich gebildeten codierten kryptographisch verarbeiteten Nachricht CVB, die an die Benutzereinheit XS übertragen wird, durchgeführt wurden, wird die codierte kryptographisch bearbeitete Nachricht CVB von der mindestens einen Benutzereinheit XS oder von der zweiten Sicherungscomputereinheit SC2 empfangen 701.

Unter Verwendung des zur Codierung verwendeten Transportprotokolls wird die codierte kryptographisch bearbeitete Nachricht CVB in der Transportprotokollschicht TP der Benutzereinheit oder der zweiten Sicherungscomputereinheit SC2 "ausgepackt", also decodiert 702.

Damit wird eine decodierte kryptographisch verarbeitete Nachricht DVB gebildet, die nun der Sicherungsschicht SL, die auch auf der Seite der Benutzereinheit XS bzw. der zweiten Sicherungscomputereinheit SC2 vorgesehen ist, zugeführt. In der Sicherheitsschicht SL wird die decodierte kryptographisch bearbeitete Nachricht DVB den jeweils inversen kryptographischen Verfahren unterzogen 703. Invers bedeutet in diesem Zusammenhang invers zu den kryptographischen Verfahren, die in der Sicherheitsschicht des Klienten XC bzw. der ersten Sicherungscomputereinheit SC1 auf die decodierte Nachricht DB angewendet wurden.

Das Ergebnis dieser kryptographischen Bearbeitung ist eine invers kryptographisch bearbeitete Nachricht DEB, die nun wiederum der Transportprotokollschicht TP zugeführt wird, wo sie auch wieder codiert wird 704.

Die daraus entstandene codierte invers kryptographisch bearbeitete Nachricht CEB wird wiederum der Transportprotokollschicht TP zugeführt und dort decodiert 705.

Die resultierende Nachricht wird nunmehr dem eigentlichen Server XS, also der Benutzereinheit XS, zugeführt und dort weiterverarbeitet. Es ist selbstverständlich in einer Variante des Verfahrens auch möglich, direkt die invers kryptographisch bearbeitete Nachricht DEB weiter zu verarbeiten.

Die prinzipiell gleiche Weiterbildung des Verfahrens gemäß Patentanspruch 4 wie die im vorigen beschriebene Weiterbildung für das Verfahren gemäß Patentanspruch 1 ist in **Fig.** 10 dargestellt sowie die zur Durchführung des Verfahrens gemäß Patentanspruch 5 benötigte Anordnung in **Fig.** 17.

Bei dieser Weiterbildung wird wiederum davon ausgegangen, daß die in **Fig.** 9 beschriebenen Verfahrensschritte bis zur Codierung der kryptographisch bearbeiteten Anforderung VA und deren Übertragung an das Programm P durchgeführt wurden.

Die übertragene codierte kryptographisch bearbeitete Anforderung CVA wird von dem Programm P oder von der ersten Sicherungscomputereinheit SC1 empfangen 1001.

In einem weiteren Schritt 1002 wird unter Verwendung des Transportprotokolls wiederum die codierte kryptographisch bearbeitete Anforderung CVA "ausgepackt", also decodiert in der Transportprotokollschicht

TP.

Weiter wird die daraus resultierte decodierte kryptographisch bearbeitete Anforderung DVA in der Sicherheitsschicht SL, der sie zugeführt wurde, der zu dem eingesetzten kryptographischen Verfahren inversen kryptographischen Verarbeitung unterzogen 1003.

Die resultierende invers kryptographisch bearbeitete Anforderung DEA wird wiederum in der Transportprotokollschicht TP codiert 1004.

Anschließend wird sie in der Transportprotokollschicht TP wiederum decodiert 1005 und dem Programm P zugeführt. Dort wird die eigentliche Anforderung A weiterverarbeitet.

Wiederum ist es ebenso möglich, direkt die decodierte invers kryptographisch bearbeitete Anforderung DEA dem Programm P zuzuführen und dort weiterzuverarbeiten.

In **Fig.** 11 ist ein weiteres Verfahren, das ebenso auf der gemeinsamen erfinderischen Idee der im vorigen beschriebenen Verfahren basiert, beschrieben.

Hierbei wird jedoch vorausgesetzt, daß es möglich ist, direkt eine Sicherheitsschicht SL zwischen dem Klienten XC und der Transportprotokollschicht TP einzufügen. Hieraus ergibt sich nunmehr nicht mehr die Notwendigkeit auf der Seite des Klienten XC die Transportprotokollschicht TP zweimal zu "durchlaufen".

Dies ist in der Anordnung von **Fig.** 17 dargestellt.

Hierbei wird wiederum von dem Programm P die Nachricht B gebildet 1101. Die Nachricht B wird jedoch diesmal direkt in der Sicherheitsschicht S- einen kryptographischen Verfahren unterzogen VB, 1102. Die resultierende kryptographisch bearbeitete Nachricht VB wird der Transportprotokollschicht TP zugeführt, wo sie codiert wird 1103.

Die codierte kryptographisch bearbeitete Nachricht CVB wird an die Benutzereinheit XS übertragen 1104, dort von der Benutzereinheit XS oder der zweiten Sicherungscomputereinheit SC2 empfangen 1105, in der dort vorgesehenen Transportprotokollschicht TP decodiert zu der decodierten kryptographisch bearbeiteten Nachricht DVB 1106.

Diese wird der Sicherheitsschicht SL zugeführt und dort dem bzw. den inversen kryptographischen Verfahren unterzogen 1107.

In zwei letzten Schritten wird die invers kryptographisch bearbeitete Nachricht DEB in der Transportprotokollschicht TP wiederum codiert 1108 und in einem letzten Schritt decodiert 1109.

Die daraus resultierende Nachricht B wird dem Server XS zugeführt und weiter verarbeitet.

Die Sicherheitsschicht SL ist in **Fig.** 4 dargestellt für den Fall, daß es möglich ist, die Sicherheitsschicht SL zwischen die Transportschicht TP und die Bibliotheksroutinen XL einzufügen.

Hierbei werden für das spezielle Beispiel, das jedoch die Allgemeingültigkeit in keinster Weise einschränkt, ungesicherte read, write, readv, writev connect und accept Nachrichten durch in der Sicherheitsschicht SL vorgesehene kryptographische Verfahren "abgesichert". Dies erfolgt durch Anwendung der vorgesehenen kryptographischen Verfahren auf die jeweilige Nachricht B bzw. Anforderung A. Die durch die Sicherheitsschicht SL "gesicherten" Nachrichten sind mit einem Stern * in **Fig.** 4 gekennzeichnet.

Die beschriebene kryptographische Sicherung der Kommunikation einer Anwendung mit einem Fenstersystem über ein Netz setzt einerseits den Austausch kryptographischer Schlüssel voraus und beruht andererseits auf einer wechselseitigen Authentikation der beiden Kommunikationspartner.

Zu dieser Authentikation können asymmetrische, kryptographische Verfahren mitsamt Zertifikaten, die öffentliche Schlüssel enthalten, vorteilhaft eingesetzt werden. Durch eine geeignete Definition der Identitätsmerkmale in dem Zertifikat ist es möglich, Dienste wie Anwendungen oder Fensterdienstprogramme über die reine Rechneradresse im Netz hinaus zu identifizieren und zu authentisieren. Solche über die Netzadresse hinausgehenden Identitätsmerkmale zur Differenzierung verschiedener Anwendungsprogramme eines Rechners können z. B. der Name des Dienstebesitzers auf einem Mehrbenutzersystem sein.

Die wechselseitige Authentikation und der Schlüsselaustausch werden in einer Initialisierungsphase zum Aufbau der sicheren Verbindung realisiert.

In einer Weiterbildung des erfindungsgemäßen Verfahrens ist es vorteilhaft, auf der Seite des Fensterdienstprogrammes, also der Benutzereinheit XC eine Zugriffskontrolle auf Basis der authentifizierten Identität des Programmes P durchzuführen. Da die authentifizierte Identifikationsinformation über die Rechneradresse des Programmes P hinausgehen kann, kann eine Zugriffskontrolle zwischen verschiedenen Programmes P eines Rechners unterscheiden und somit den Verbindungsaufbau steuern.

Eine vorteilhafte Anwendung der beschriebenen Sicherungsverfahren findet sich beim Austausch von Anwendungsdaten zwischen einem Programmes P und einem Fensterdienstprogramm, also einer Benutzereinheit XC, wobei zwischen beiden nur eine nicht vertrauenswürdige Netzverbindung geschaltet werden kann.

Dieses Szenario ist von besonderer Bedeutung für die oben beschriebenen CSCW-Systeme, die Application Sharing realisieren. Hierbei befinden sich die beteiligten Fensterdienstprogramme der Benutzereinheiten XC oftmals in verschiedenen Firmennetzen und können Daten mit der Anwendung bzw. der Application Sharing Komponente nur über öffentliche Netze austauschen.

Mit dem Betreiben des bekannten Fenstersystems sind erhebliche Sicherheitsprobleme verbunden, welche in [6], [7] beschrieben werden. Aufgrund des erheblichen Risikopotentials, welches mit dem aus [1] bekannten Fenstersystem verbunden ist, lassen es die Betreiber von Firmennetzen in der Regel nicht zu, daß solche Fensterdienstprogramme mit Anwendungen außerhalb des Firmennetzes zusammenarbeiten. Dies dient dem Schutz firmeninterner Informationen und Datenbestände. Dieser Schutz wird durch sogenannte Firewalls am Netzübergang zwischen internem Netz und externen Netzen realisiert. Diese verhindern durch eine Filterung von Datenpaketen auf Transportsystemebene, daß externe Anwendungsprogramme auf interne Fensterdienstprogramme zugreifen.

Diese üblichen Vorkehrungen verhindern aber die Nutzung synchroner CSCW-Systeme, die darauf beruhen, daß Nutzer an unterschiedlichen Standorten und in verschiedenen Firmen gemeinsam durch ein synchrones CSCW-System kooperieren und gemeinsam mit Anwendungsprogrammen arbeiten.

Auf Basis des beschriebenen Sicherungsverfahren für Anwendungsdaten läßt sich ein Programm für ein Firewall konstruieren, welches es ermöglicht, interne Fensterdienstprogramme auf sichere Weise mit externen Anwendungsprogrammen kommunizieren zu lassen:

Dieses spezielle Programm beruht einerseits auf der beschriebenen Sicherheitserweiterung zum Schutz von

Anwendungsdaten in Fenstersystemen und andererseits auf einer Durchschaltekomponente für Anwendungsdaten. Die Durchschaltekomponente kann direkt aus der Application Sharing Komponente ASC abgeleitet werden, da hierzu die Anforderung des Multiplexens und Demultiplexens entfällt.

Diese beiden Komponenten (Sicherheitsdienstprogramm und Durchschaltekomponente) bilden ein spezielles Firewall-Sicherheitsdienstprogramm durch welches es möglich wird, von einem externen Anwendungsprogramm eine spezifische Authentikation zu verlangen, sowie es einer Zugriffskontrolle zu unterziehen, bevor die Durchschaltekomponente die Verbindung zu dem internen Fensterdienstprogramm herstellt und anschließend die Verbindung durchstellt. Der nachfolgende Datenaustausch zwischen dem externen Anwendungsprogramm und dem Firewall-Sicherheitsdienstprogramm wird durch kryptographische Mechanismen geschützt.

Durch das Betreiben von Paketfiltern im Firewall können externe Anwendungsprogramme gezwungen werden, zunächst die Verbindung zu dem beschriebenen Sicherheitsdienstprogramm aufzunehmen.

Das entsprechende Verfahren unter Berücksichtigung des "Rollentauschs" zwischen Programm und Benutzereinheit XS, also für die Anforderung A, ist in **Fig.** 12 sowie die zu deren Durchführung benötigte Anordnung in **Fig.** 18 dargestellt.

Hierbei wird natürlich angenommen, daß die Sicherheitsschicht SL auf der Seite der Benutzereinheit XS zwischen die Benutzereinheit XS und die Transportprotokollschicht TP eingefügt werden kann.

Unter dieser Annahme wird also von der Benutzereinheit XS die Anforderung A gebildet 1201. Diese wird direkt in der Sicherheitsschicht SL dem kryptographischen Verfahren VA unterzogen 1202.

Die kryptographisch bearbeitete Anforderung VA wird in der Transportprotokollschicht TP codiert 1203 und im Anschluß daran wird die codierte kryptographisch bearbeitete Anforderung CVA an das Programm P übertragen 1204.

Dort wird sie von dem Programm P oder von der ersten Sicherungscomputereinheit SC1 empfangen 1205. In einer auch dort vorgesehenen Transportprotokollschicht TP wird sie nunmehr decodiert zur decodierten kryptographisch bearbeiteten Anforderung DVA 1206.

In der Sicherheitsschicht SL der sie in einem weiteren Schritt zugeführt wird, wird die decodierte kryptographische Anforderung dem inversen kryptographischen Verfahren unterzogen 1207. Die daraus resultierende invers kryptographisch bearbeitete Anforderung DEA wird in der Transportprotokollschicht TP wiederum "eingepackt", also codiert 1208.

Die codierte inverse kryptographische bearbeitete Anforderung CEA wird in der Transportprotokollschicht TP in einem weiteren Schritt wiederum decodiert 1209 und die resultierende Anforderung A, die nunmehr kryptographisch "abgesichert" ist, wird dem Programm zugeführt und von dem Programm P weiter verwendet.

Verschiedene Möglichkeiten zur Realisierung der zu verwendenden kryptographischen Verfahren in der Sicherheitsschicht SL sind in **Fig.** 8 dargestellt.

Zum einen ist es möglich, Verschlüsselungsverfahren 81 in der Sicherheitsschicht SL anzuwenden. Damit wird eine Vertraulichkeit bzw. Integrität der ausgetauschten Nachrichten B bzw. Anforderungen A erreicht.

Ferner ist es vorgesehen, in der Sicherheitsschicht SL auch Authentikationsmechanismen 82 zu verwenden. Diese erlauben es, Identitätsangaben der Kommunikationspartner im Netz zu verifizieren. Diese Authentikationsmechanismen haben besondere Bedeutung in Zusammenhang zum Beispiel des Transport Control Protocols (TCP), oder auch des User Datagramm Protocols (UDP), da diese keinerlei Authentikationsmechanismen für Sender und Empfänger aufweisen.

Auch die Realisierung von Zugriffskontrollmechanismen 83, die auf den Authentikationsverfahren beruhen, bietet zusätzlichen Schutz des Zugangs zu dem Fensterdienstprogramm in einem Client-Server-Fenster-System.

Die im vorherigen beschriebenen Verfahren können natürlich auch auf Mehrbenutzersysteme sehr vorteilhaft angewendet werden.

Wie das [1] beschriebene Client-Server-Fenstersystem erweitert werden kann zu einem Mehrbenutzersystem ist beispielsweise beschrieben in [3], [4], [5].

Die daraus resultierende Situation mit einer zusätzliche Multiplexerkomponente ASC und mehreren Benutzereinheiten XSi, wobei ein Index i jede Benutzereinheit XSi eindeutig identifiziert und eine natürliche Zahl im Bereich von 1 bis n ist, ist in **Fig.** 2 dargestellt.

Hierbei werden in bekannter Weise die Anforderungen Ai von den einzelnen Benutzereinheiten XSi zusammengeführt und die Nachricht B wird an die einzelnen Benutzereinheiten XSi als Kopien der Nachricht Bi gesendet.

Die erfindungsgemäßen Verfahren werden in diesem Zusammenhang natürlich für jede einzelne Verbindung zwischen dem Klienten XC und jeder Benutzereinheit XSi einzeln durchgeführt.

Detaillierter ist diese "Mehrbenutzer-Umgebung" noch in **Fig.** 3 dargestellt. In dieser Realisierung entsprechen die Anforderungen Ai sogenannten Xrequests und die Nachrichten Bi den sogenannten Xreplies, Xevents, Xerrors. Die Anwendung ANW greift über Systemaufrufe SC auf Systemressourcen SR zu.

In einer Weiterbildung des Verfahrens ist es vorteilhaft, zu Beginn des Verfahrens eine Initialisierungsphase vorzusehen, in der beispielsweise ein Schlüsselaustausch sowie eine beidseitige Authentikation zwischen einer Benutzereinheit XS der Benutzereinheiten XSi und dem Programm P durchgeführt wird.

Hierbei sind dem Fachmann unterschiedlichste Verfahren zum Schlüsselaustausch bekannt. Als Beispiel einer Initialisierungsphase, die in dem erfindungsgemäßen Verfahren eingesetzt werden kann, wird folgendes Vorgehen vorgeschlagen.

Das im folgende beschriebene Verfahren zum Schlüsselaustausch wird allgemein zwischen dem Klienten XC und einer Benutzereinheit XS durchgeführt. Die Multiplexerkomponente ASC ist in diesem Zusammenhang als eine spezielle Komponente des Klienten XC zu betrachten.

Unter der Annahme, daß die Multiplexerkomponente ASC ein Anwendungszertifikat besitzt und die Benutzereinheiten, also die Server XSi, jeweils ein Benutzerzertifikat besitzen, die jeweils eindeutig den Benutzereinheiten zugeordnet sind, wird dann von der Multiplexerkomponente ASC eine erste Zufallszahl erzeugt.

Nachdem eine Transportverbindung zwischen der Multiplexerkomponente ASC und dem jeweiligen Server XSi aufgebaut wurde, wird von der Multiplexerkomponente ASC eine erste Verhandlungsnachricht an die Benutzereinheit gesendet, die mindestens folgende

Komponenten aufweist:

- das Programmzertifikat,
- die erste Zufallszahl,
- einen ersten Vorschlag für ein im weiteren zu verwendende kryptographische Verfahren, und
- eine digitale Unterschrift, die mindestens über die erste Zufallszahl sowie den ersten Vorschlag gebildet wird.

Die erste Verhandlungsnachricht wird von der jeweiligen Benutzereinheit, also dem Server XSi, empfangen.

Von der Benutzereinheit XSi wird das Programmzertifikat auf Korrektheit überprüft.

Ferner wird die digitale Unterschrift überprüft.

Falls die Überprüfung des Programmzertifikats und der digitalen Unterschrift ein positives Ergebnis liefert, wird in der Benutzereinheit XSi weiterhin überprüft, ob die vorgeschlagenen kryptographischen Algorithmen die in der ersten Verhandlungsnachricht vorgeschlagen wurden, im weiteren zur Sicherung der Übertragung verwendet werden können.

Wenn die Benutzereinheit XSi die vorgeschlagenen kryptographischen Algorithmen nicht unterstützen kann, wird von der Benutzereinheit, also dem Server XSi, ein zweiter Vorschlag in einer zweiten Vorschlagsnachricht gebildet und an die Multiplexerkomponente ASC gesendet. Der zweite Vorschlag weist kryptographische Verfahren auf, die die Benutzereinheit XSi unterstützt. Diese werden nunmehr der Multiplexerkomponente ASC als im weiteren Verfahren zu verwendende kryptographische Verfahren für diese logische Verbindung zwischen der Multiplexerkomponente und der Benutzereinheit XSi vorgeschlagen.

Die zweite Vorschlagsnachricht weist mindestens folgende Komponenten auf:

- das Benutzerzertifikat des jeweiligen Servers XSi,
- eine zweite Zufallszahl, die von der Benutzereinheit XSi selbst erzeugt wurde,
- den zweiten Vorschlag,
- eine digitale Unterschrift, die jeweils mindestens über die erste Zufallszahl, die zweite Zufallszahl sowie den zweiten Vorschlag gebildet werden.

Die zweite Vorschlagsnachricht wird an die Multiplexerkomponente ASC gesendet.

Für den Fall, daß die in dem ersten Vorschlag angegebenen kryptographischen Algorithmen von dem Benutzereinheit XSi unterstützt werden, wird von dem Benutzereinheit XSi eine Bestätigungsnachricht gebildet und an die Multiplexerkomponente ASC gesendet.

Die Bestätigungsnachricht weist mindestens folgende Komponenten auf:

- das Benutzerzertifikat,
- die zweite Zufallszahl,
- eine positive Bestätigung, und
- eine digitale Unterschrift, die jeweils mindestens über die erste Zufallszahl, die zweite Zufallszahl, und die positive Bestätigung gebildet werden.

Die Bestätigungsnachricht wird an die Multiplexerkomponente ASC gesendet.

Von der Multiplexerkomponente ASC wird die Verhandlungsnachricht oder die Bestätigungsnachricht empfangen und es wird in der Multiplexerkomponente ASC geprüft, ob das Benutzerzertifikat sowie die digitale Unterschrift korrekt sind.

Weiterhin wird von der Multiplexerkomponente ASC für den Fall, daß die Überprüfung ein positives Ergebnis liefert und die empfangene Nachricht die Bestätigungsnachricht war, ein erster Sitzungsschlüssel unter Berücksichtigung der vereinbarten kryptographischen Algorithmen für eine folgende Nutzdatenübertragungsphase erzeugt.

Aus dem ersten Sitzungsschlüssel wird eine erste Sitzungsschlüsselnachricht gebildet und an die Benutzereinheit XSi gesendet, die mindestens folgende Komponenten aufweist:

- den mit einem öffentlichen Schlüssel des Servers XSi verschlüsselten ersten Sitzungsschlüssel,
- eine Spezifikation der zu verwendenden kryptographischen Verfahren,
- eine mindestens über die erste Zufallszahl, die zweite Zufallszahl, den ersten Sitzungsschlüssel gebildete digitale Unterschrift sowie die Spezifikation der zu verwendenden kryptographischen Verfahren.

Wurde von der Multiplexerkomponente ASC die zweite Verhandlungsnachricht empfangen, und die Überprüfung des Benutzerzertifikats und der digitalen Unterschrift oder des Hash-Werts der zweiten Verhandlungsnachricht hat ein positives Ergebnis geliefert, wird in der Multiplexerkomponente ASC geprüft, ob die in der zweiten Verhandlungsnachricht vorgeschlagenen kryptographischen Algorithmen zur Durchführung der weiteren kryptographischen Verfahren von der Multiplexerkomponente ASC unterstützt werden.

Werden die vorgeschlagenen kryptographischen Verfahren von der Multiplexerkomponente ASC unterstützt, wird ein erster Sitzungsschlüssel unter Berücksichtigung der vereinbarten kryptographischen Algorithmen für die folgende Nutzdatenübertragungsphase erzeugt.

Weiterhin wird, wie im vorigen beschrieben wurde, eine erste Sitzungsschlüsselnachricht unter Verwendung des ersten Sitzungsschlüssels an die Multiplexerkomponente ASC gesendet.

Diese im vorigen beschriebene Vorgehensweise zum "Aushandeln" der zu verwendenden kryptographischen Verfahren wird solange wiederholt, bis sowohl die Benutzereinheit XSi als auch die Multiplexerkomponente ASC zuletzt vorgeschlagenen kryptographischen Verfahren akzeptieren.

In der Benutzereinheit XSi wird der erste Sitzungsschlüssel unter Verwendung eines privaten Schlüssels der Benutzereinheit XSi ermittelt. Ferner wird die digitale Unterschrift der ersten Sitzungsschlüsselnachricht überprüft.

Außerdem wird für den Fall, daß die Überprüfung der digitalen Unterschrift ein positives Ergebnis lieferte, eine zweite Sitzungsschlüsselnachricht gebildet unter Verwendung eines zweiten Sitzungsschlüssels, der von der Benutzereinheit XSi gebildet wird.

Die zweite Sitzungsschlüsselnachricht weist mindestens folgende Komponenten auf:

- den mit einem öffentlichen Programmschlüssel der Multiplexerkomponente ASC verschlüsselten zweiten Sitzungsschlüssel, und
- eine mindestens über die erste Zufallszahl, die zweite Zufallszahl, den zweiten Sitzungsschlüssel

gebildete Digitale Unterschrift oder einen über dieselben Komponenten gebildeten Hash-Wert.

Von der Multiplexerkomponente ASC wird die zweite Sitzungsschlüsselnachricht empfangen und der zweite Sitzungsschlüssel ermittelt. Die digitale Unterschrift oder der Hash-Wert der zweiten Sitzungsschlüsselnachricht wird überprüft.

Lieferte die Prüfung der digitalen Unterschrift ein positives Ergebnis, werden die ausgetauschten Sitzungsschlüssel in der folgenden Nutzdatenübertragungsphase zur Verschlüsselung der Nutzdaten verwendet. Dabei verwendet jede beteiligte Instanz den Sitzungsschlüssel, der von ihr selbst generiert wurde zum Senden von Nutzdaten, während der empfangene Sitzungsschlüssel ausschließlich zum Empfangen von Nutzdaten verwendet wird.

Weitere kryptographische Verfahren zum Schlüsselaustausch bzw. zur Bildung des Sitzungsschlüssels für die Nutzdatenverschlüsselung sind im Rahmen des erfindungsgemäßen Verfahrens ohne Einschränkungen einsetzbar.

Die erfindungsgemäßen Verfahren können sehr vorteilhaft in folgendem Szenario eingesetzt werden.

In vielen privaten Netzen werden zwischen vernetzten Rechnern sehr vertrauliche Informationen untereinander ausgetauscht. Hierbei ist meistens das private Netz selbst sehr gut abgesichert gegen die Außenwelt, beispielsweise durch sogenannte Firewalls [6].

Wenn nun ein an das jeweils abgesicherte private Netz angeschlossene Rechner mit einem sich außerhalb dieses Netzes, nur über einen unsicheren Kanal erreichbaren Rechner kommunizieren möchte, beispielsweise einem nur über das Internet IN erreichbaren Rechner, besteht bisher ein großes Problem darin, daß bei dem auf [1] basierenden Client-Server-Fenster-Systemen keine sichere Kommunikation möglich ist.

Insbesondere ergibt sich das Problem, daß über Fensterdienstprogramme andere Anwendungen angegriffen werden können. Um das Ausspähen interner Informationen zu verhindern, ist es in Firmennetzen in der Regel nicht erlaubt, ein Fensterdienstprogramm außerhalb des Firmennetzes zu betreiben. Diese allgemein übliche Beschränkung behindert insbesondere synchrone CSCW-Systeme, die auf Application Sharing beruhen.

Diese Probleme sind beispielsweise in [6], [7] ausführlich geschildert.

Es muß sich bei diesem Problem nicht unbedingt um eine über ein lokales Netz übergreifende Kommunikation handeln, sondern es kann sich beispielsweise auch um ein abgesichertes Corporate Network CN handeln, bei dem ein Kommunikationspartner mit einem anderen Kommunikationspartner einer anderen Firma über den Rechner beispielsweise in einem CSCW-System kommunizieren möchte.

Durch die erfindungsgemäßen Verfahren ist es nunmehr möglich, bei Einsätzen dieser Verfahren in einem Firewall SC1, SC2 des lokalen Netzes bzw. des Corporate Networks CN, wobei eben der Firewall jeweils als erste Sicherungscomputereinheit SC1 bzw. als zweite Sicherungscomputereinheit SC2 anzusehen ist (vgl. **Fig. 3**).

In dieser Schrift wurden folgende Veröffentlichungen zitiert:

[1] R. Scheifler et al, The X Window System, ACM Transactions on Graphics, Vol. 5, No. 2, S. 79 bis 109, April 1986

[2] S. Garfinkel et al, Practical UNIX Security, O'Reilly & Associates, Inc., ISBN 0-937175-72-2, S. 221—253, 1991

[3] H. Abdel-Wahab et al, Issues, Problems and Solutions in Sharing X Clients on Multiple Displays, Internetworking: Research and Experience, Vol. 5, S. 1 bis 15, 1994

[6] D. Garfinkel et al, HP Shared X: A Tool for Real-Time Collaboration, Hewlett-Packard Journal, S. 23 bis 36, April 1994

[5] J. Baldeschwieler et al, A Survey of X Protocol Multiplexors, Swiss Federal Institute of Technology, Computer Engineering and Networks Laboratory (TIK), ETH-Zentrum, Zürich, 1993

[6] S. Bellovin et al, Network Firewalls, IEEE Communications Magazin, S. 50 bis 57, September 1994

[7] G. Treese et al, X Through the Firewall, and Other Application Relays, Summer Usenix, 1993, 21. bis 25. Juni, Cincinnati, S. 87 bis 98, 1993

Patentansprüche

1. Verfahren zur kryptographischen Sicherung der rechnergestützten digitalen Kommunikation zwischen einem Programm (P) und mindestens einer Benutzereinheit (XSi),

— bei dem von dem Programm (P) eine Nachricht (B) gebildet wird (601)

— bei dem von einer Rechnereinheit, auf der das Programm (P) verarbeitet wird, oder von einer ersten Sicherungscomputereinheit (SC1) die Nachricht (B) mit einem Transportprotokoll codiert (CB) wird (602),

— bei dem die codierte Nachricht (CB) unter Verwendung des Transportprotokolls decodiert (DB) wird (603),

— bei dem die decodierte Nachricht (DB) einem kryptographischen Verfahren (VB) unterzogen wird (604),

— bei dem die kryptographisch bearbeitete Nachricht (VB) mit dem Transportprotokoll codiert (CVB) wird (605), und

— bei dem die codierte kryptographisch bearbeitete Nachricht (CVB) an die mindestens eine Benutzereinheit (XSi) übertragen wird (606)

2. Verfahren nach Anspruch 1,

— bei dem die codierte kryptographisch bearbeitete Nachricht (CVB) von der mindestens einen Benutzereinheit (XSi) oder von einer zweiten Sicherungscomputereinheit (SC2) empfangen wird (701),

— bei dem unter Verwendung des Transportprotokolls die codierte kryptographisch bearbeitete Nachricht (CVB) decodiert (DVB) wird (702),

— bei dem die decodierte kryptographisch bearbeitete Nachricht (DVB) einer zu dem kryptographischen Verfahren inversen kryptographischen Bearbeitung (DEB) unterzogen wird (703),

— bei dem die invers kryptographisch bearbeitete Nachricht (DEB) mit dem, Transportprotokoll codiert (CEB) wird (704), und

— bei dem die codierte invers kryptographisch bearbeitete Nachricht (CEB) unter Verwendung des Transportprotokolls decodiert wird (705).

3. Verfahren nach Anspruch 1,

— bei dem die codierte kryptographisch bearbeitete Nachricht (CVB) von der mindestens einen Benutzereinheit (XSi) empfangen wird,

— bei dem unter Verwendung des Transportprotokolls die codierte kryptographisch bearbeitete Nachricht (CVB) decodiert (DVB) wird, und

— bei dem die decodierte kryptographisch bearbeitete Nachricht (DVB) einer zu dem kryptographischen Verfahren inversen kryptographischen Bearbeitung (DEB) unterzogen wird.

4. Verfahren zur kryptographischen Sicherung der rechnergestützten digitalen Kommunikation zwischen einem Programm (P) und mindestens einer Benutzereinheit (XSi),

— bei dem von einer Benutzereinheit (XSi) eine Anforderung (A) gebildet wird (901),

— bei dem von der Benutzereinheit (XSi) oder von einer zweiten Sicherungscomputereinheit (SC2) die Anforderung (A) mit einem Transportprotokoll codiert (CA) wird (902),

— bei dem die codierte Anforderung (CA) unter Verwendung des Transportprotokolls decodiert (DA) wird (903),

— bei dem die decodierte Anforderung (DA) einem kryptographischen Verfahren (VA) unterzogen wird (904),

— bei dem die kryptographisch bearbeitete Anforderung (VA) mit dem Transportprotokoll codiert (CVA) wird (905), und

— bei dem die codierte kryptographisch bearbeitete Anforderung (CVA) an das Programm (P) übertragen wird (906).

5. Verfahren nach Anspruch 4,

— bei dem die codierte kryptographisch bearbeitete Anforderung (CVA) von dem Programm (P) oder von einer ersten Sicherungscomputereinheit (SC1) empfangen wird (1001),

— bei dem unter Verwendung des Transportprotokolls die codierte kryptographisch bearbeitete Anforderung (CVA) decodiert (DVA) wird (1002),

— bei dem die decodierte kryptographisch bearbeitete Anforderung (DVA) einer zu dem kryptographischen Verfahren inversen kryptographischen Bearbeitung (DEA) unterzogen wird (1003), — bei dem die invers kryptographisch bearbeitete Anforderung (DEA) mit dem Transportprotokoll codiert (CEA) wird (1004), und

— bei dem die codierte invers kryptographisch bearbeitete Anforderung (CEA) unter Verwendung des Transportprotokolls decodiert wird (1005).

6. Verfahren nach Anspruch 4,

— bei dem die codierte kryptographisch bearbeitete Anforderung (CVA) von dem Programm (P) empfangen wird,

— bei dem unter Verwendung des Transportprotokolls die codierte kryptographisch bearbeitete Anforderung (CVA) decodiert (DVA) wird, und

— bei dem die decodierte kryptographisch bearbeitete Anforderung (DVA) einer zu dem kryptographischen Verfahren inversen kryptographischen Bearbeitung (DEA) unterzogen wird.

7. Verfahren zur kryptographischen Sicherung der rechnergestützten digitalen Kommunikation zwischen einem Programm (P) und mindestens einer Benutzereinheit (XSi),

— bei dem von dem Programm (P) eine Nachricht (B) gebildet wird (1101),

— bei dem die Nachricht (B) einem kryptographischen Verfahren (VB) unterzogen wird (1102),

— bei dem die kryptographisch bearbeitete Nachricht (VB) mit dem Transportprotokoll codiert (CVB) wird (1103),

— bei dem die codierte kryptographisch bearbeitete Nachricht (CVB) an die mindestens eine Benutzereinheit (XSi) übertragen wird (1104),

— bei dem die codierte kryptographisch bearbeitete Nachricht (CVB) von der mindestens einen Benutzereinheit (XSi) oder von einer zweiten Sicherungscomputereinheit (SC2) empfangen wird (1105),

— bei dem unter Verwendung des Transportprotokolls die codierte kryptographisch bearbeitete Nachricht (CVB) decodiert (DVB) wird (1106)

— bei dem die decodierte kryptographisch bearbeitete Nachricht (DVB) einer zu dem kryptographischen Verfahren inversen kryptographischen Bearbeitung (DEB) unterzogen wird (1107),

— bei dem die invers kryptographisch bearbeitete Nachricht (DEB) mit dem Transportprotokoll codiert (CEB) wird (1108), und

— bei dem die codierte invers kryptographisch bearbeitete Nachricht (CEB) unter Verwendung des Transportprotokolls decodiert wird (1109).

8. Verfahren zur kryptographischen Sicherung der rechnergestützten digitalen Kommunikation zwischen einem Programm (P) und mindestens einer Benutzereinheit (XSi),

— bei dem von der mindestens einen Benutzereinheit (XSi) eine Anforderung (A) gebildet wird (1201),

— bei dem die decodierte Anforderung (DA) einem kryptographischen Verfahren (VA) unterzogen wird (1202),

— bei dem von der Benutzereinheit (XSi) oder von einer zweiten Sicherungscomputereinheit (SC2) die kryptographisch bearbeitete Anforderung (A) mit einem Transportprotokoll codiert (CA) wird (1203),

— bei dem die codierte kryptographisch bearbeitete Anforderung (CVA) an das Programm (P) übertragen wird (1204),

— bei dem die codierte kryptographisch bearbeitete Anforderung (CVA) von dem Programm (P) oder von einer ersten Sicherungscomputereinheit (SC1) empfangen wird (1205),

— bei dem unter Verwendung des Transportprotokolls die codierte kryptographisch bearbeitete Anforderung (CVA) decodiert (DVA) wird (1206),

— bei dem die decodierte kryptographisch bearbeitete Anforderung (DVA) einer zu dem kryptographischen Verfahren inversen kryptographischen Bearbeitung (DEA) unterzogen wird (1207),

— bei dem die invers kryptographisch bearbeitete Anforderung (DEA) mit dem Transportprotokoll codiert (CEA) wird (1208), und

— bei dem die codierte invers kryptographisch bearbeitete Anforderung (CEA) unter Verwendung des Transportprotokolls decodiert wird (1209).

9. Verfahren nach einem der Ansprüche 1 bis 8,

— bei dem die kryptographische Bearbeitung mindestens durch eine Verschlüsselung der Anforderung (Ai) realisiert ist (81), und

— bei dem die inverse kryptographische Bearbeitung mindestens durch eine Entschlüsselung der Anforderung (Ai) realisiert ist.

10. Verfahren nach einem der Ansprüche 1 bis 9,

— bei dem die kryptographische Bearbeitung mindestens durch Authentikationsmechanismen für die Anforderung (Ai) realisiert ist (82), und

— bei dem die inverse kryptographische Bearbeitung mindestens durch inverse Authentikationsmechanismen für die Anforderung (Ai) realisiert ist.

11. Verfahren nach einem der Ansprüche 1 bis 10,

— bei dem die kryptographische Bearbeitung mindestens durch Zugriffskontrollmechanismen für die Anforderung (Ai) realisiert ist (83), und

— bei dem die inverse kryptographische Bearbeitung mindestens durch inverse Zugriffskontrollmechanismen für die Anforderung (Ai) realisiert ist.

12. Verfahren nach einem der Ansprüche 1 bis 11, bei dem zu Beginn des Verfahrens eine kryptographische Initialisierungsphase mit Bildung eines Sitzungsschlüssels für jede Verbindung einer Benutzereinheit (XSi) mit dem Programm (P) durchgeführt wird.

Hierzu 13 Seite(n) Zeichnungen

- Leerseite -

Nummer: DE 195 48 387 C1
Int. Cl.[6]: H 04 L 9/00
Veröffentlichungstag: 30. Januar 1997

FIG 1

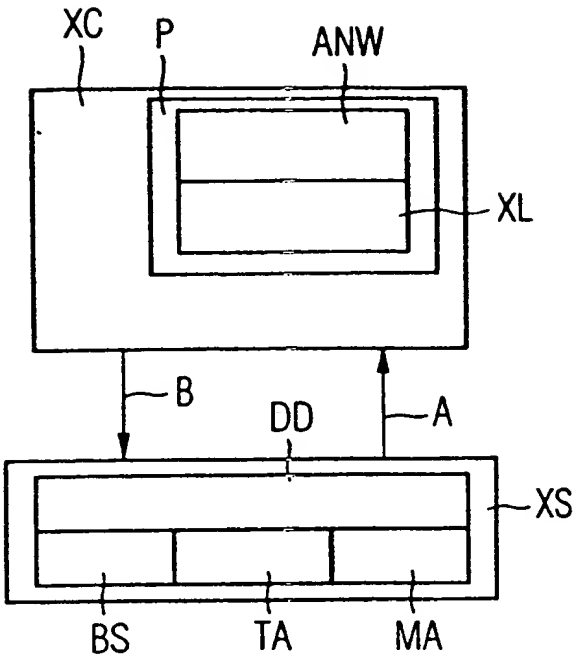
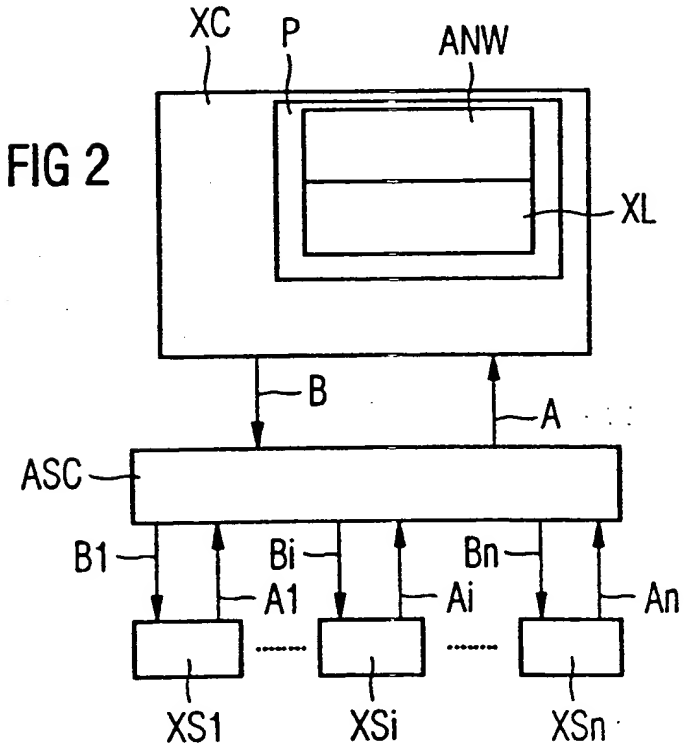


FIG 2



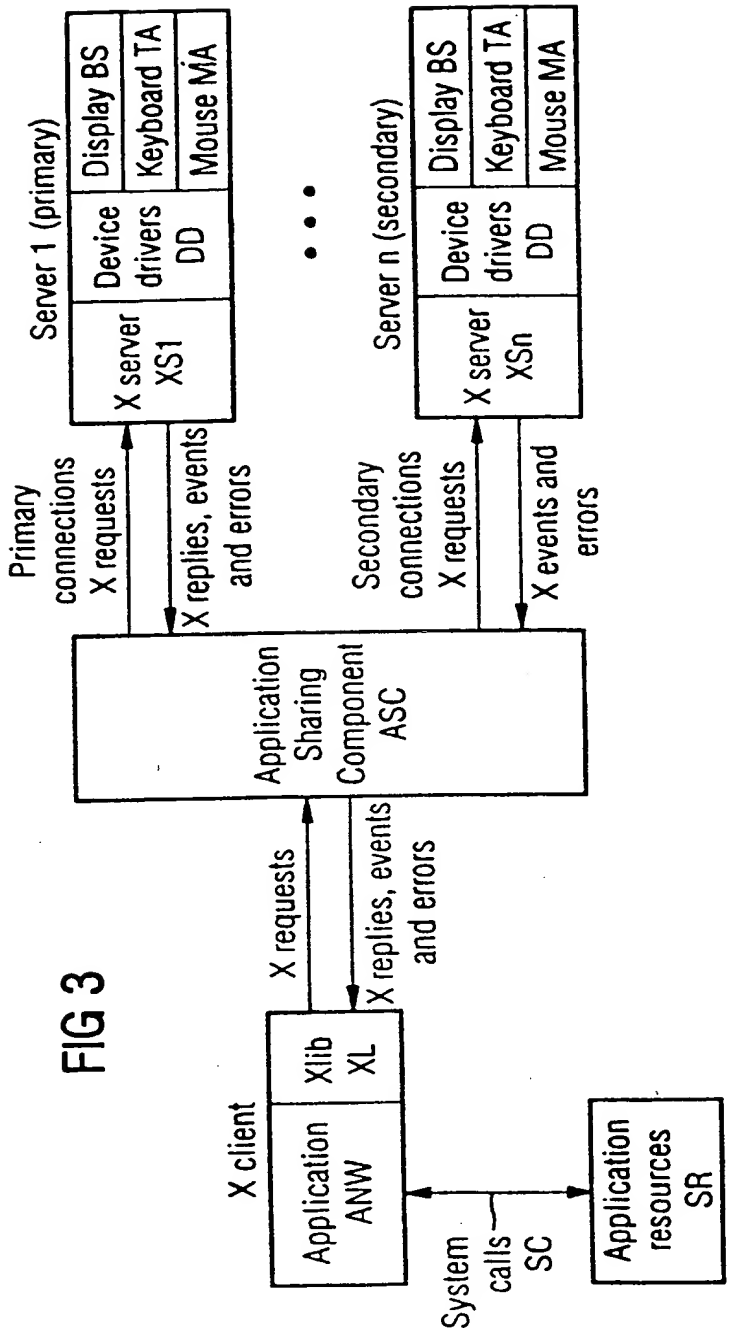


FIG 3

Nummer: DE 195 48 367 C1
Int. Cl.[6]: H 04 L 9/00
Veröffentlichungstag: 30. Januar 1997

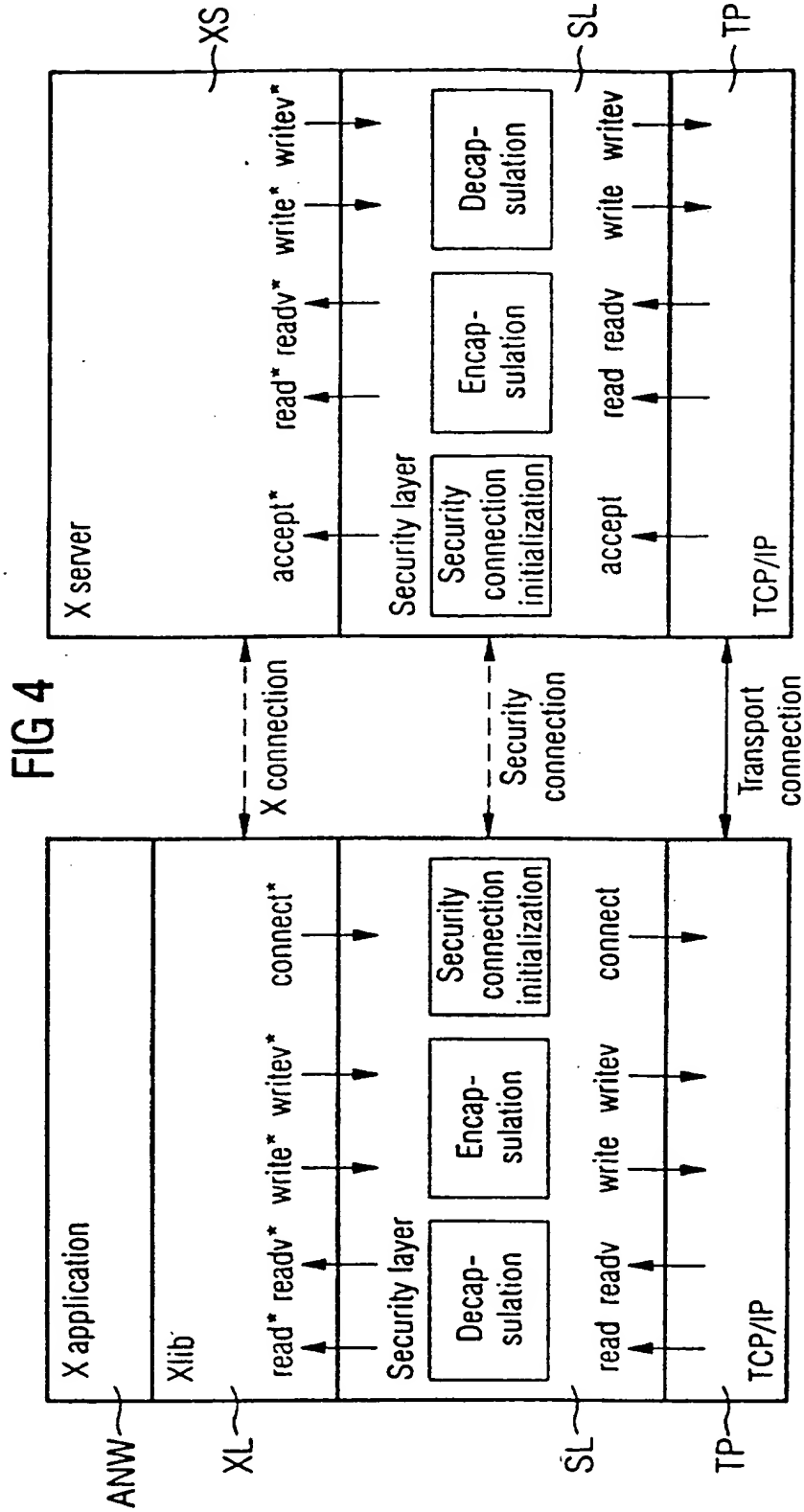
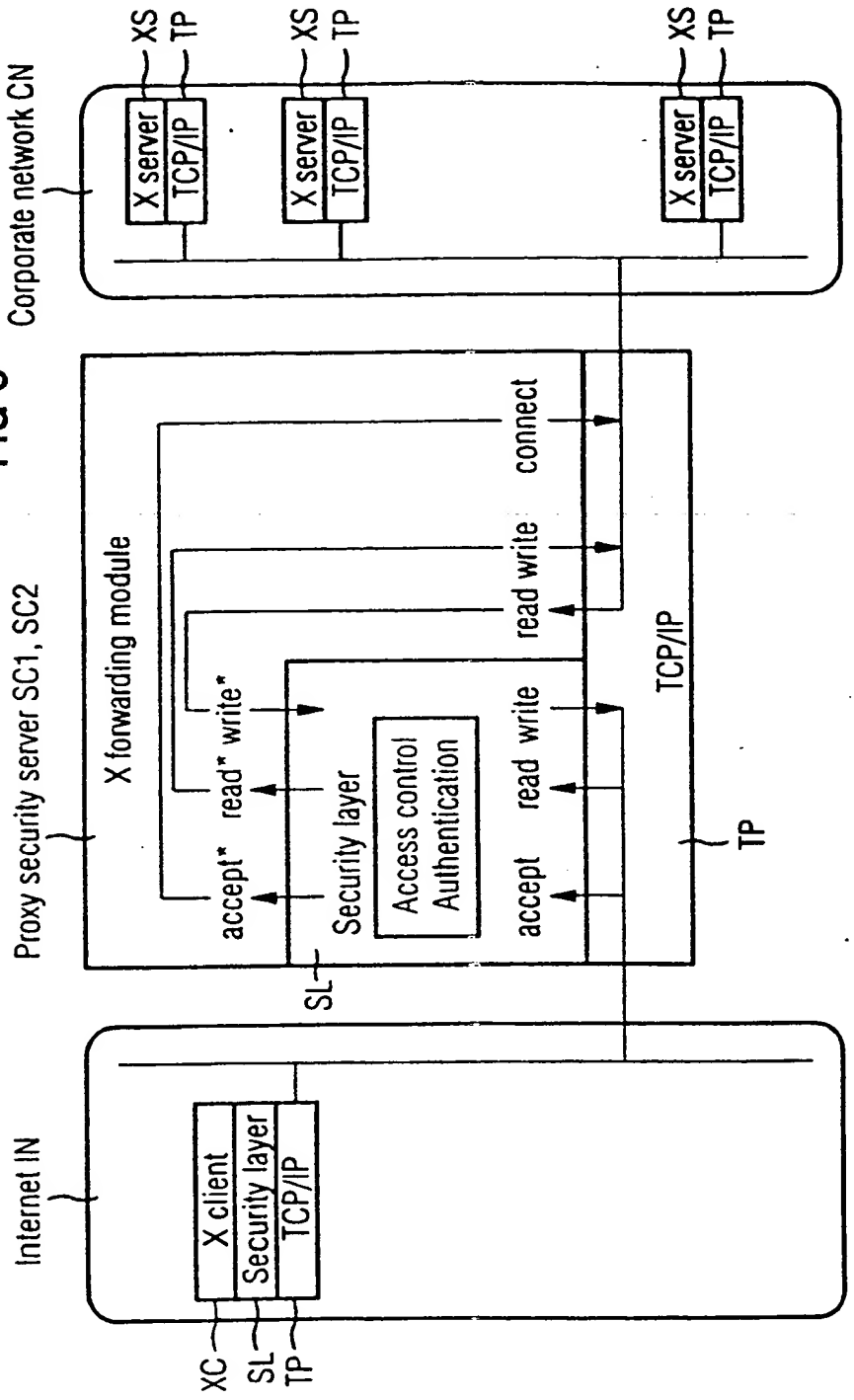


FIG 4

FIG 5



Nummer: DE 195 48 387 C1
Int. Cl.[6]: H 04 L 9/00
Veröffentlichungstag: 30. Januar 1997

FIG 6

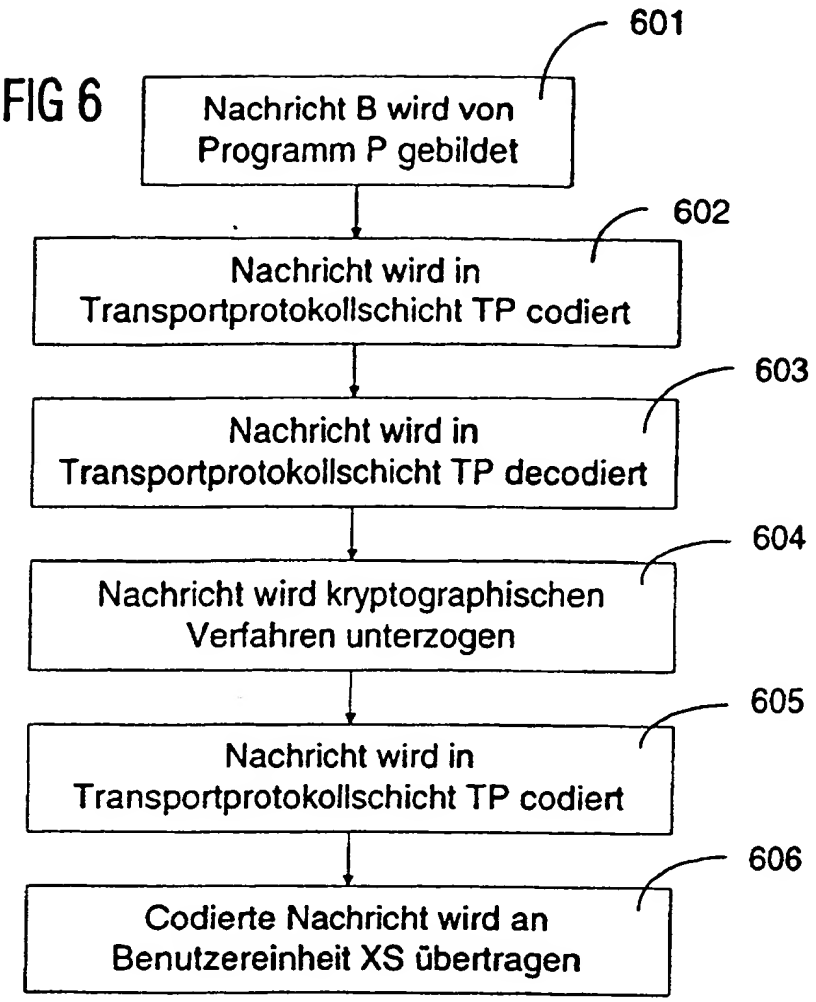


FIG 8

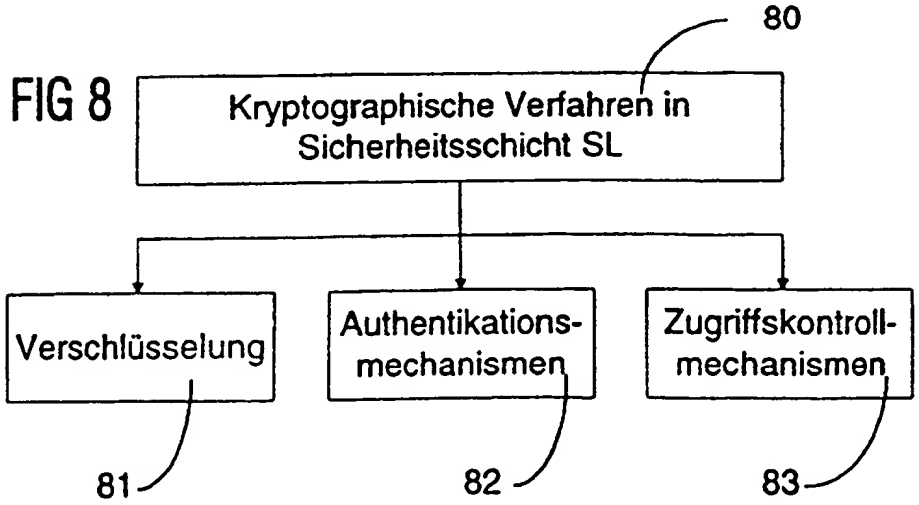
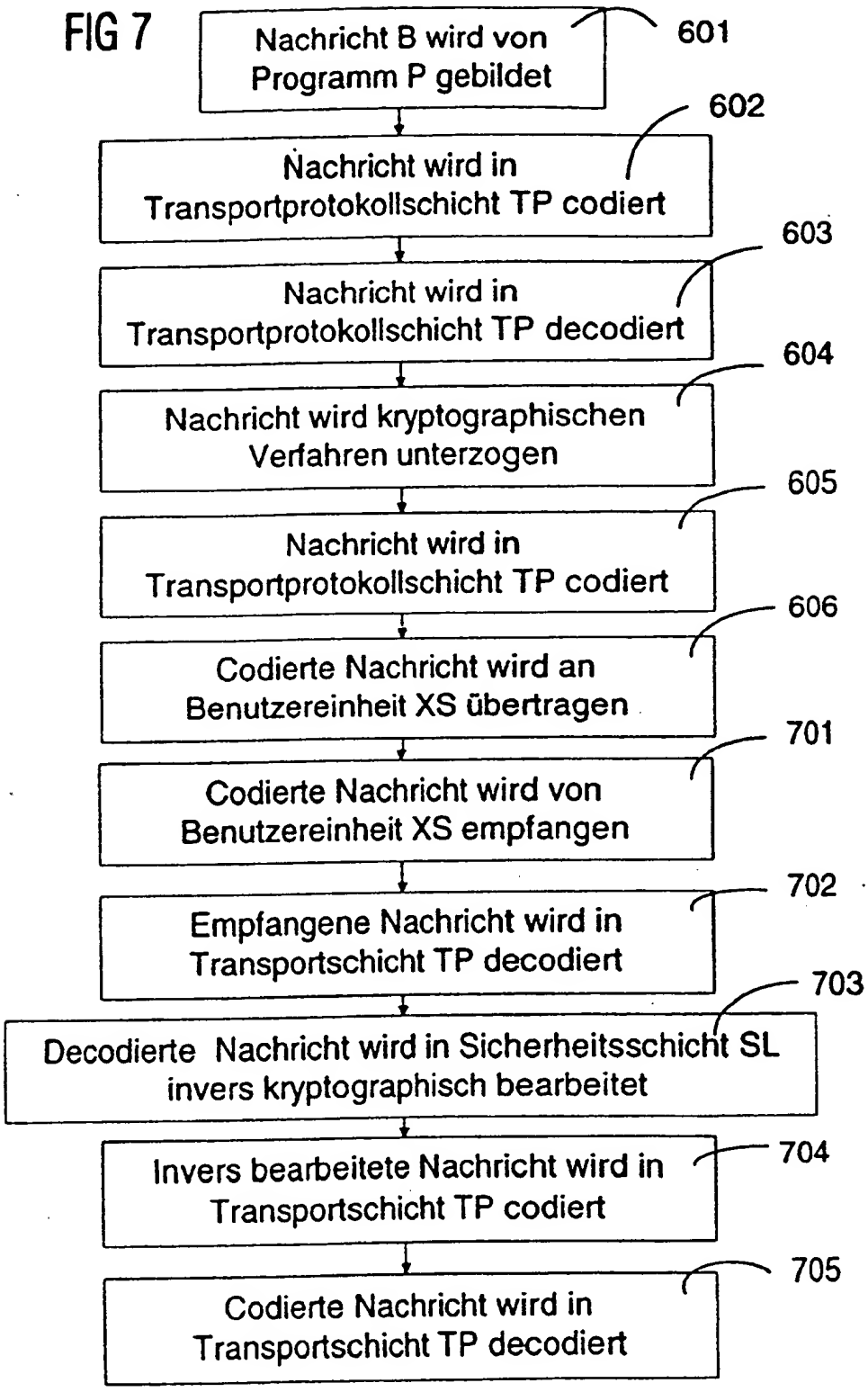


FIG 7



Nummer: DE 195 48 387 C1
Int. Cl.[6]: H 04 L 9/00
Veröffentlichungstag: 30. Januar 1997

FIG 9

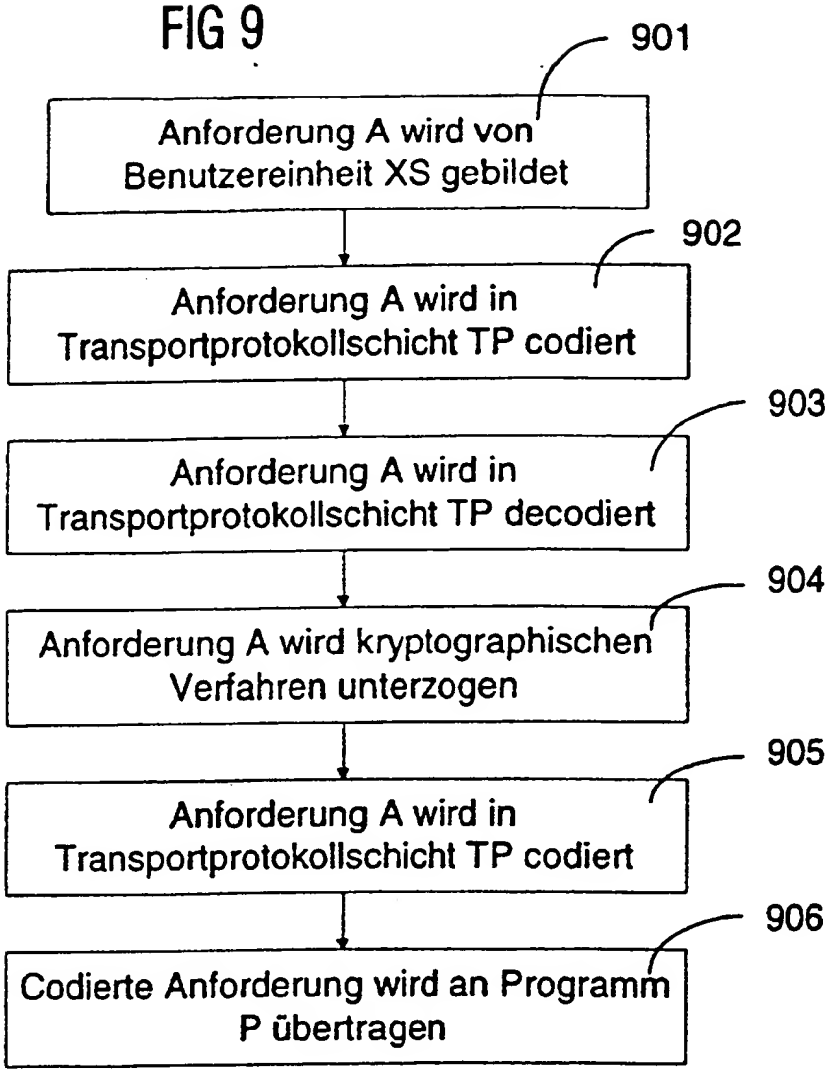
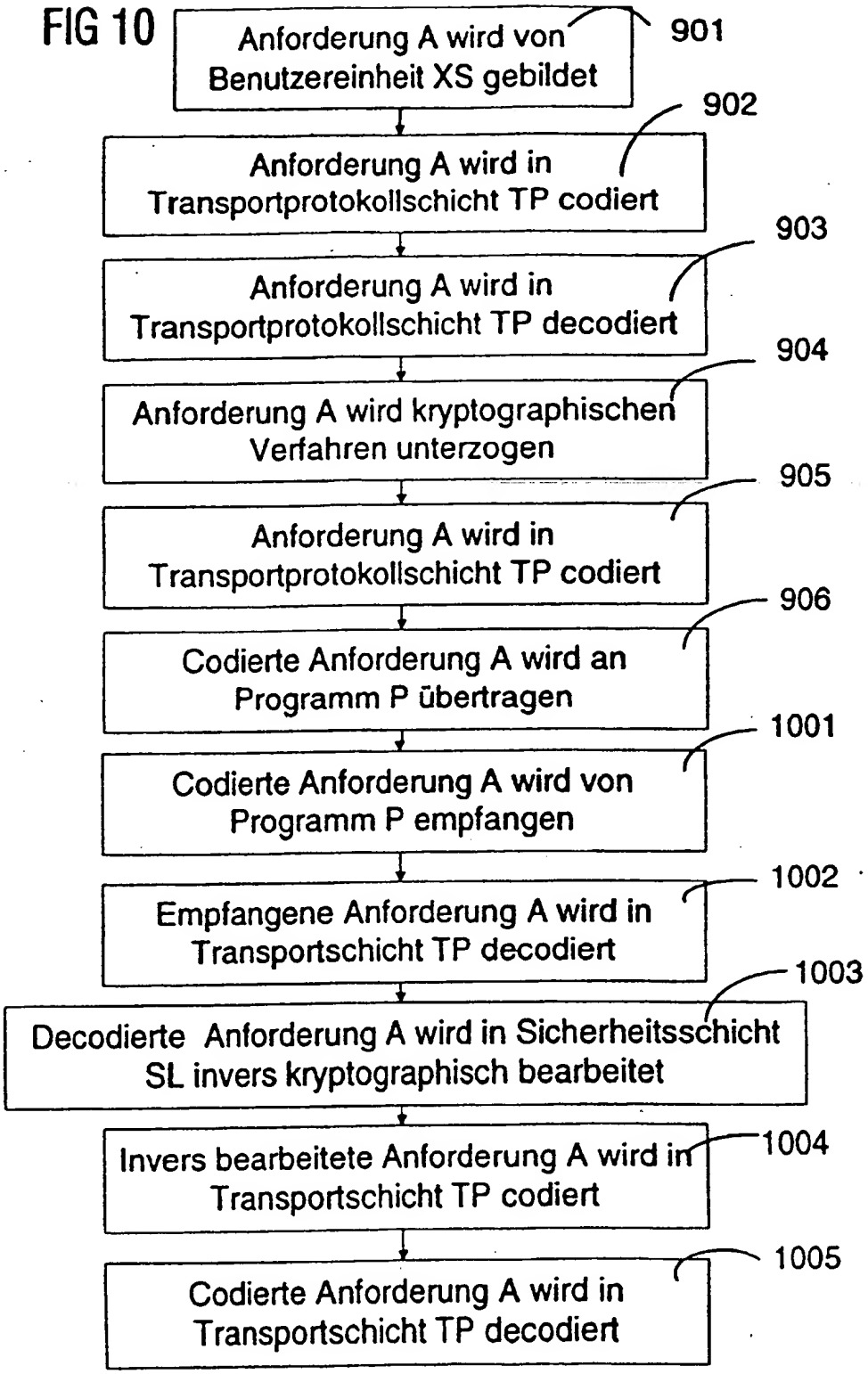


FIG 10



Nummer: DE 195 48 387 C1
Int. Cl.[6]: H 04 L 9/00
Veröffentlichungstag: 30. Januar 1997

FIG 11

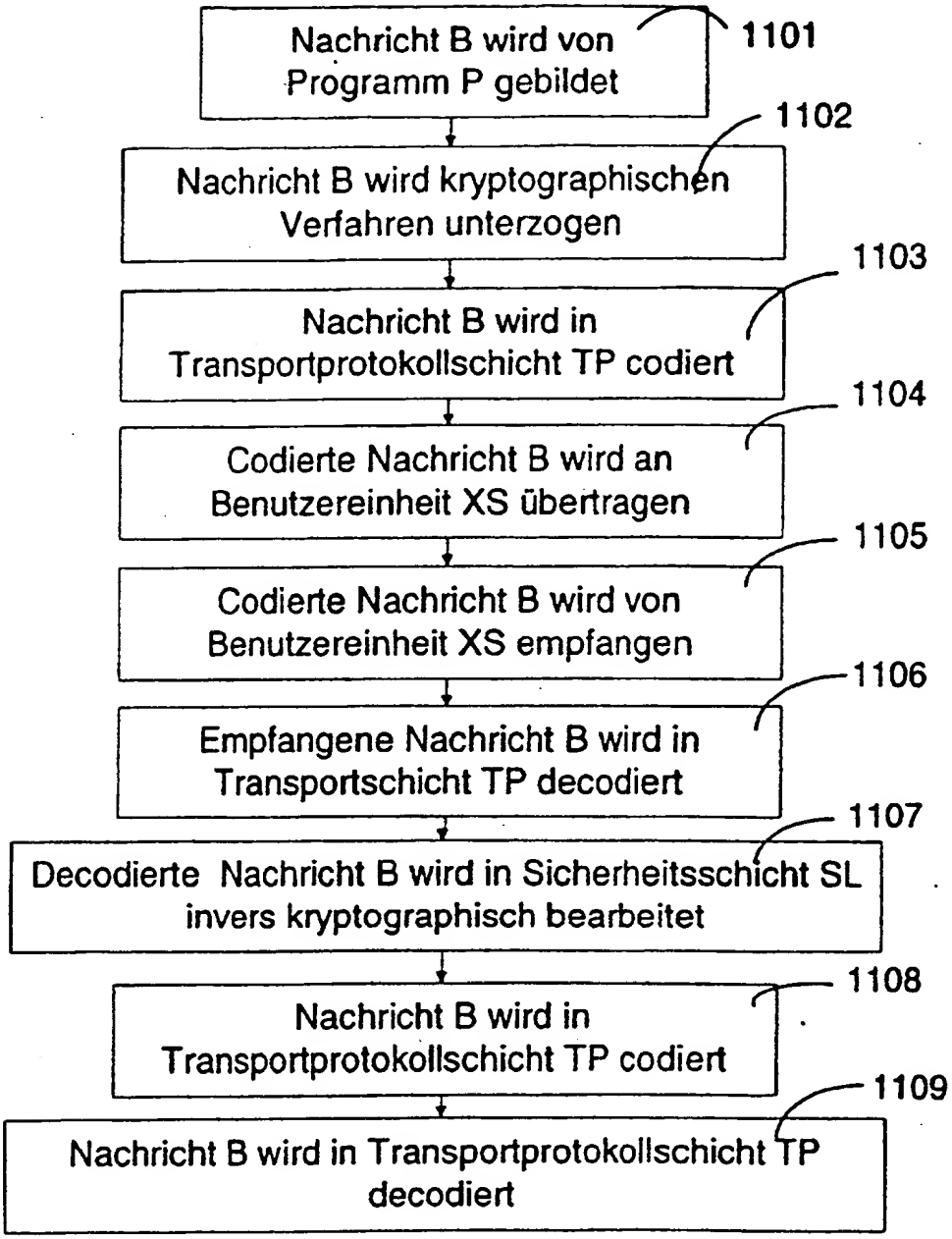
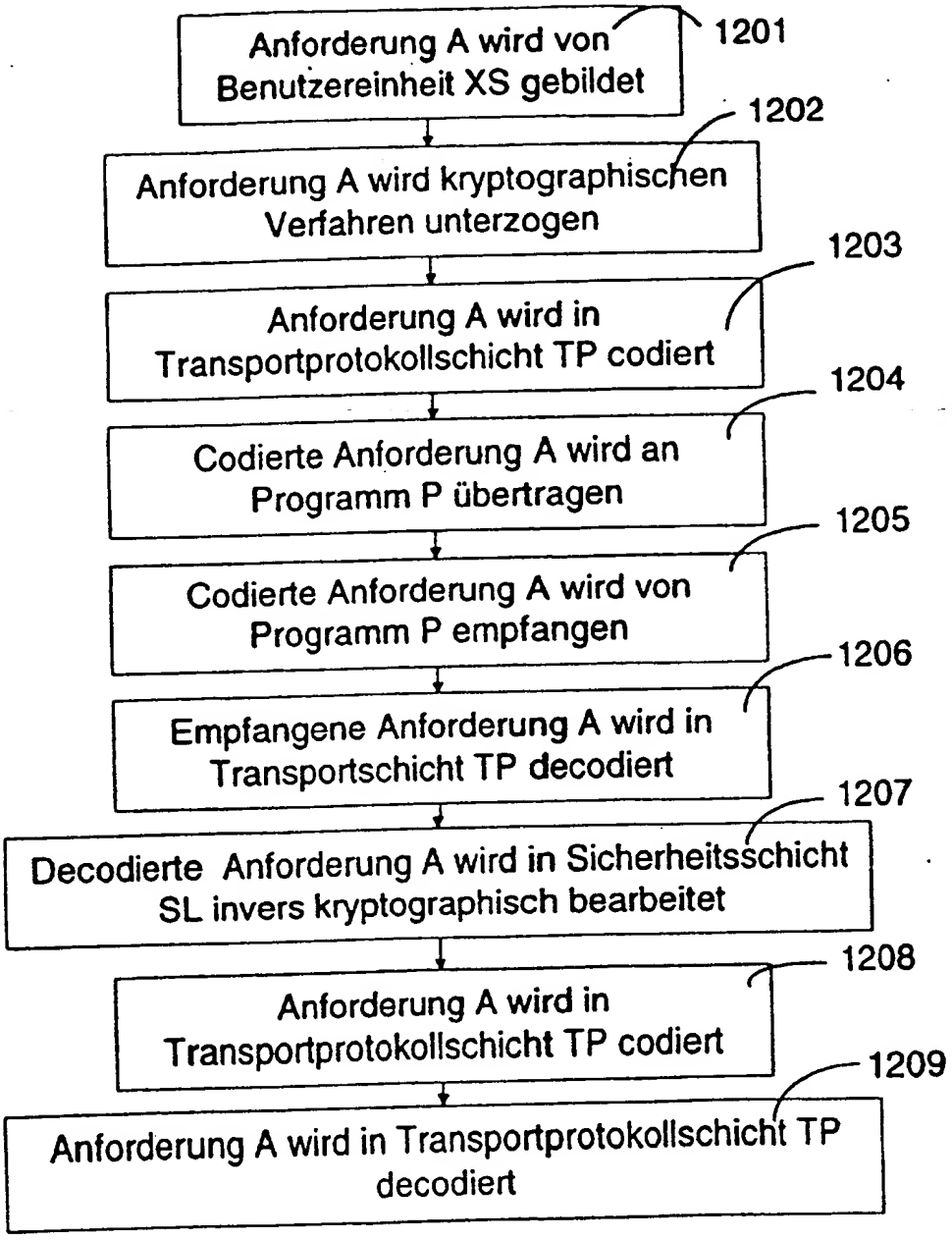


FIG 12



Nummer: DE 196 48 387 C1
Int. Cl.[6]: H 04 L 9/00
Veröffentlichungstag: 30. Januar 1997

FIG 13

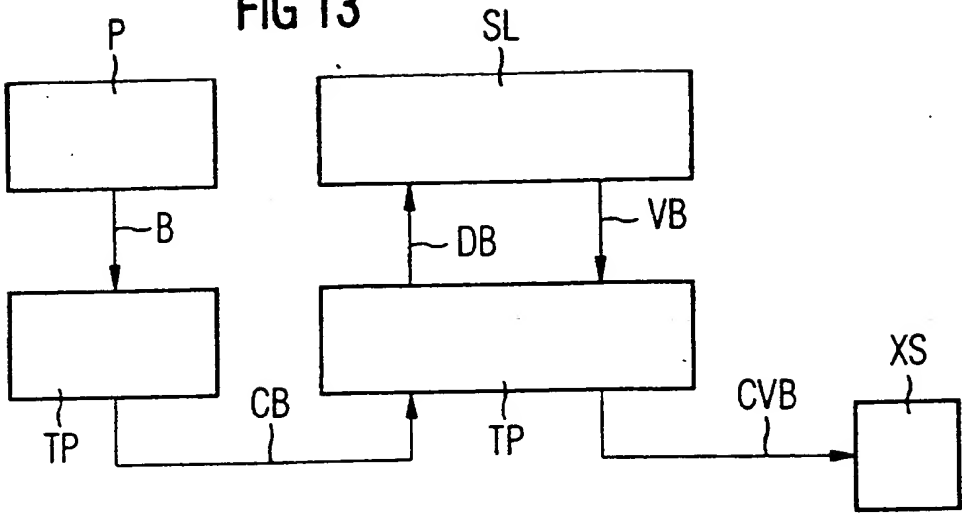


FIG 14

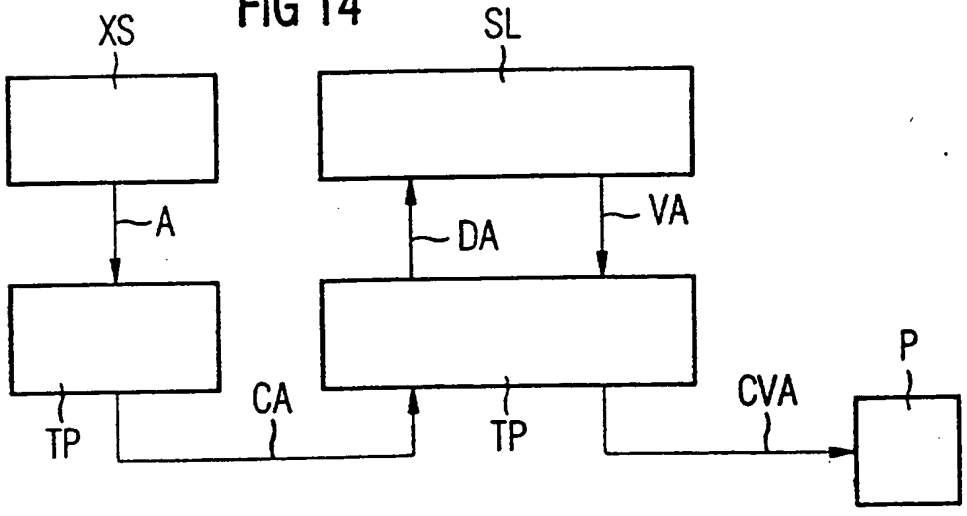


FIG 15

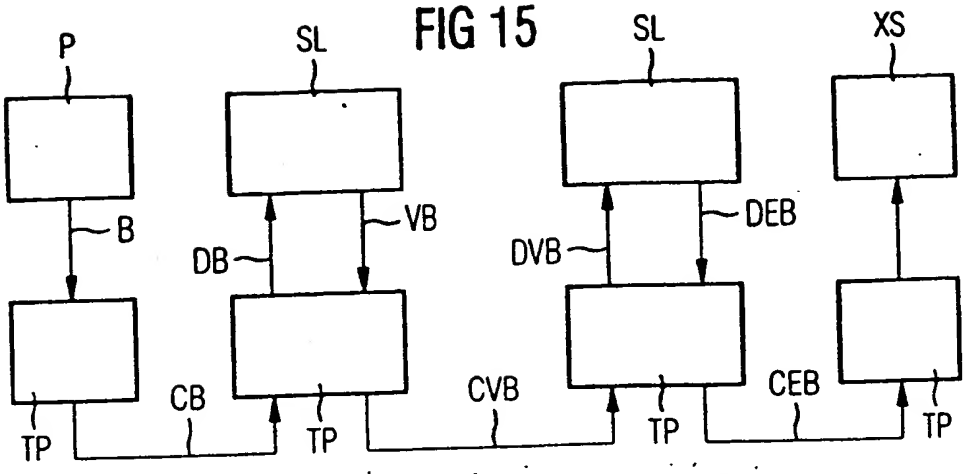
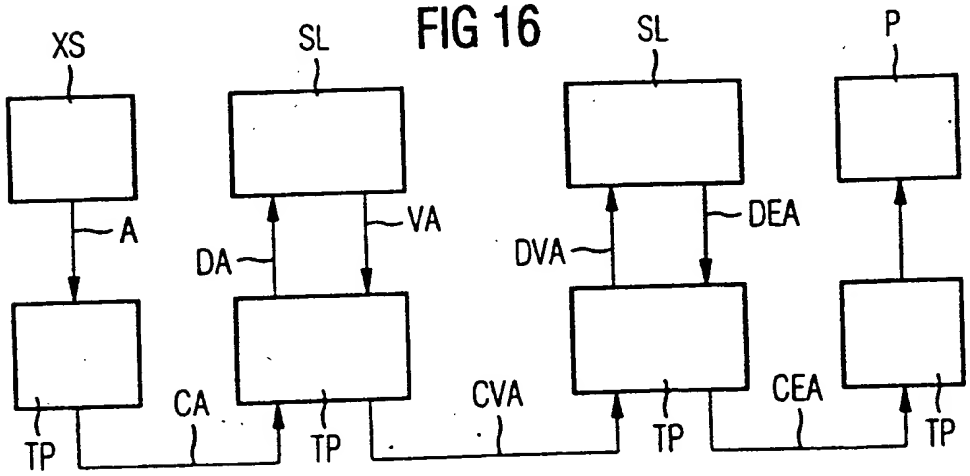


FIG 16



Nummer: DE 195 48 387 C1
Int. Cl.[6]: H 04 L 9/00
Veröffentlichungstag: 30. Januar 1997

FIG 17

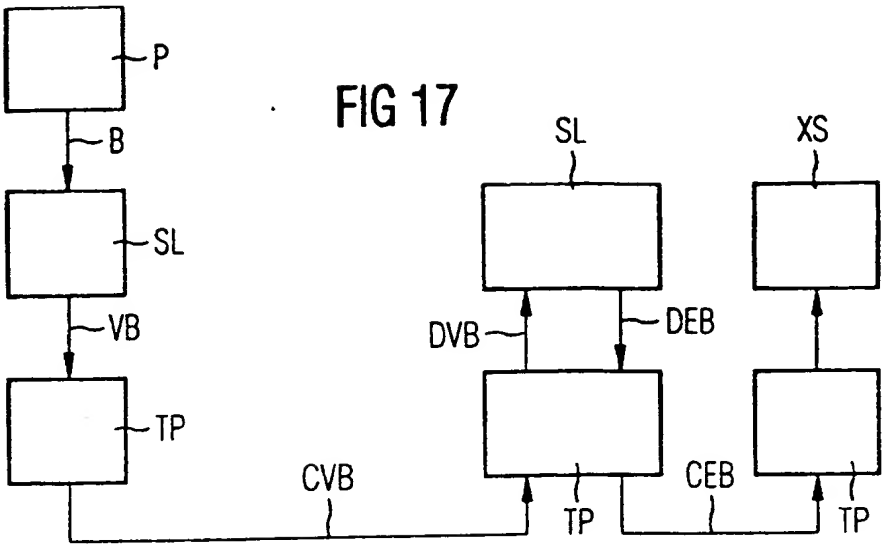
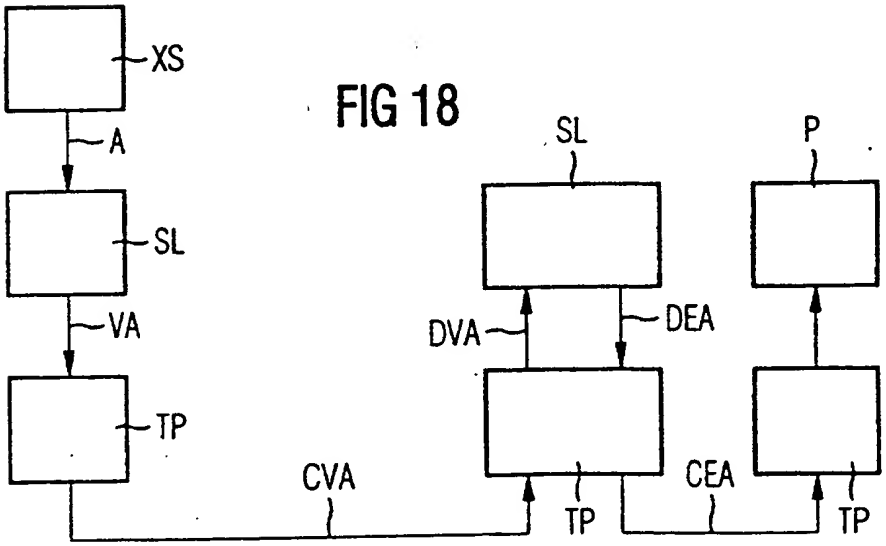


FIG 18



PTO 03-4961

German

Document No. DE 195 48 387 C1

Process for Cryptographic Securing of Computer-Assisted Digital Communication Between a Program and at Least One User Unit

[Verfahren zur kryptographischen Sicherung der rechnergestützten digitalen Kommunikation zwischen einem Programm und mindestens einer Benutzereinheit]

Dr. Oliver Pfaff

UNITED STATES PATENT AND TRADEMARK OFFICE

Washington, D.C. August 2003

Translated by: Schreiber Translations, Inc.

Country	:	Germany
Document No.	:	DE 195 48 387 C1
Document Type	:	Patent Application Laid Open to Inspection
Language	:	German
Inventor	:	Dr. Oliver Pfaff
Applicant	:	Siemens AG
IPC	:	H 04 L 9/00, G 06 F 12/14, G 06 F 13/42
Application Date	:	December 22, 1995
Publication Date	:	January 30, 1997
Foreign Language Title	:	Verfahren zur kryptographischen Sicherung der rechnergestützten digitalen Kommunikation zwischen einem Programm und mindestens einer Benutzereinheit
English Language Title	:	Process for Cryptographic Securing of Computer-Assisted Digital Communication Between a Program and at Least One User Unit

/1[1]

Specification

This invention relates to a process for the cryptographic securing of communication in so-called Client-Server window systems with an open network interface. An example of such Client-Server window systems is described in [1].

By additionally using a so-called Application Sharing Component, which is used to multiplex the requests of user units addressed to the program and to demultiplex communications from the program, for example, event messages, responses or error messages, one can achieve the common use of a Standard Single-User Application environments that can be located in different places.

The common processing of a program is referred to as Application Sharing.

But to achieve reliable communication of confidential data, the Client-Server window system, described in [1], must be expanded by cryptographic characteristics.

This is particularly important in communication between various enterprises. Here is what that means: In case of a communication between computers where one computer is in the basically secured, confidence-worthy, so-called Corporate

[1] Numbers in the margin indicate pagination in the foreign text.

Network of an enterprise and other computers that [are] in a common synchronously distributed working environment of several cross-linked computers in a so-called Computer System Cooperated Work System (CWCW systems) that implements Application Sharing, can be achieved only via an unsecured channel, which means that secure communication is no longer guaranteed.

Such CSCW systems are based on the possibility of processing a standard single-user application (Standard Single User Application) together with other units at one particular point in time.

The information exchanged between the computers can be of special significance, for example, it may involve confidential business information, design specifications, financial transactions or medical data that are exchanged via the unsecured channel.

This is why it is necessary to ensure a certain degree of security also for these transactions that involve application data.

In many commercial systems that are based on the Client-Service window system described in [1], direct integration of cryptographic characteristics is not possible.

The problem behind the invention thus is to provide a process for the cryptographic securing of computer-assisted

digital communication between a program and at least one user unit.

The problem is solved by the process according to Claim 1, the process according to Claim 4, the process according to Claim 7 as well as the process according to Claim 8.

In the process according to Claim 1, the program forms a communication that is coded for at transport protocol. Directly after encoding using the transport protocol, the encoded communication is again decoded and the decoded communication is subjected to the cryptographic process. Then the request is again encoded with the transport protocol and is transmitted to at least one user unit. In this case, the program and the user unit can be in one or also on various computers.

In the process according to Claim 4, one basically takes the same steps with the difference that this time a request is formed in a user unit and that the additional steps described above are also taken there. To complete the procedure, the encoded cryptographically processed request is transmitted to the program in this case.

In the process according to Claim 7, one starts with the following situation: On the side of the program, an expansion of the Client-Server window system is possible by means of security mechanisms of the most varied kind, which will be described below. For this case, the above-described process

steps will be performed in a user unit starting with the formation of a communication in the program only after reception of the requests that were cryptographically processed in the inserted security layer on the side of the program. In other words, the inverse cryptographic processes are performed there in order to process the communications and this process step is characterized by a prior decoding with the transport protocol and subsequent encoding with the transport protocol.

The process according to Claim 8 basically features the same process steps as the process according to Claim 7 with the following difference: Here a request is formulated by a user unit and transferred to the program. The placed where the individual process steps occur and the places where the process steps of Claim 7 are performed are simply switched around in this case.

Advantageous developments of the invention-based process will result from the subclaims.

It is advantageous by way of cryptographic processing to provide at least encoding of the request. That ensures the confidentiality of the exchanged data.

It is also advantageous by way of cryptographic processing of the request to provide integrity and authentication mechanisms; this means that one can in each case guarantee then

that the received communication actually comes from the sender, who is also listed as sender in the communication.

As a further development of the invention-based process, it is also advantageous to provide access control mechanisms as cryptographic processes in order thus to make sure that really only those requests are carried out which also have the entitlement for implementation.

It is also advantageous prior to the start of the process during an initialization phase, for example, to exchange between the program and at least one user unit the cryptographic codes that are employed to carry out the individual cryptographic processes.

The processes are used advantageously during data exchange between communication partners that [takes place] across the boundaries of a Corporate Network, which is secured by means of cryptographic processes via an unsecured channel in a so-called Firewall. /2

By virtue of this manner of use, it is no longer necessary as in the past in handling an intended communication beyond Corporate Network boundaries to uncouple the computer used for the communication from the entire network of the Corporate Network in order thus not to endanger the entire Corporate Network in case of possible attacks via the unsecured communication channel.

The figures show some exemplary embodiments, which will be explained in greater detail below.

Fig. 1 shows the general principle of a Client-Server window system;

Fig. 2 shows the general principle of a Client-Server window system in a "multiuser environment";

Fig. 3 shows an arrangement that describes the multiuser environment in a more detailed fashion;

Fig. 4 is a basic block diagram describing the insertion of a security layer between the Client-Server window system and the transport protocol;

Fig. 5 shows an arrangement basically illustrating how the invention-based process can be used in a Firewall to secure the communication beyond Corporate Network boundaries;

Fig. 6 is a flow chart illustrating the process steps of the process according to Claim 1;

Fig. 7 is a flow chart illustrating the process steps of the process according to Claim 2;

Fig. 8 is a block diagram describing the individual possibilities for carrying out the security-specific processing of the request or the inverse security-specific processing of the request;

Fig. 9 is a flow chart illustrating the individual process steps of the process according to Claim 4;

Fig. 10 is a flow chart illustrating the steps of the process according to Claim 5;

Fig. 11 is a flow chart illustrating the steps of the process according to Claim 7;

Fig. 12 is a flow chart illustrating the steps of the process according to Claim 8;

Fig. 13 is a block diagram describing the individual components needed to implement the process according to Claim 1 and the communication exchange [procedure];

Fig. 14 is a block diagram describing the individual components needed to implement the process according to Claim 4 and the communication exchange [procedure];

Fig. 15 is a block diagram describing the individual components needed to implement the process according to Claim 2 and the communication exchange [procedure];

Fig. 16 is a block diagram describing the individual components needed to implement the process according to Claim 5 and the communication exchange [procedure];

Fig. 17 is a block diagram describing the individual components needed to implement the process according to Claim 7 and the communication exchange [procedure];

Fig. 18 is a block diagram describing the individual components needed to implement the process according to Claim 8 and the communication exchange [procedure].

The invention will be explained in greater detail with reference to Claims 1 to 18.

Fig. 1 shows a user environment that occurs, for example, in a Client-Server window system that is described in [1].

This arrangement displays at least the following components:

- a user unit XS, hereafter also referred to as Server XS, which again has the following components:

- at least one driver unit DD which facilitates coupling between additional peripheral components with a client XC, described below,

- a display screen unit BS,

- a keyboard TA,

- a mouse MA,

- the client XC that displays at least the following components:

- a quantity of library routines XL as well as

- an application ANW.

Display screen unit BS, keyboard TA, mouse MA and possibly other peripheral units can be present additionally along with the peripheral components described above, which are coupled to client XC via the corresponding driver units DD.

The quantity of the library routines XL of client XC forms the interface between application ANW, for example, a text

processing program or also a table calculation program or all other known applications ANW and the user unit XS.

Together, the library routines XL as well as the application ANW form one program P.

Although in this exemplary embodiment we describe in each case only one application ANW or one program P, one naturally can supply several applications ANW and thus several clients XC on one computer unit that performs this particular application ANW.

This request, illustrated in Fig. 1, is thus only a very simple, basic example for the routing of the communication of a client XC with the server XS such as it is carried out via the known Client-Server window system described in [1].

Server XS transmits a request A to client XC. As a result, actions are triggered in client XC, for example, in application ANW.

Request A, for example, can be an input on keyboard TA that via the driver units DD is "translated" into request A and is transmitted to client XC.

Application ANW, for example, a text processing program or a calculation program, a symbol program and similar programs can now accept the input and, for example, include a new letter in the text data file.

To make sure that this change in the text data file can also be illustrated on the display screen BS in a response B, in this case, for example, an illustration communication, is transmitted to display screen unit BS by means of which a change in the display screen illustrated is requested. /3

One disadvantage inherent in many commercial systems that work according to this principle resides above all in the fact that a direct integration of needed security mechanisms into the Client-Server window system is often impossible.

That, it so happens, would require direct interrupt into the interface between the library routines XL and the transport protocols. The latter, it just so happens, are often not accessible to the user.

Fig. 6 shows a flow chart with individual process steps involved in the invention-based process according to Claim 1. The arrangement, needed to implement this process, is described in Fig. 13.

Program P formulates communication B in a first step 601.

A new communication is formed from communication B in a transport protocol layer TP in that communication B is "embedded" into the transport protocol format, in other words, it is coded 602, CB.

An overview of the various transport protocols can be found in [2]. The invention-based processes are independent of the special particular transport protocol that is being used.

Either on the same computer unit on which program P runs or on a separately provided first security computer unit SC1, which is coupled to the computer via a secure channel, the coded communication CB is decoded 603, DB in the transport protocol layer TP that is provided there.

The decoded communication DB is now routed to a security layer SL in which it is subjected 604 to various randomly predetermined cryptographic procedures.

A cryptographically processed communication VB, formed by cryptographic processing, is now again encoded 605 in the transport protocol layer TP, as a result of which, a coded cryptographically processed communication CVB is formed.

The coded cryptographically processed communication CVB is transferred in a last step 606 to the user unit XS, in other words, to the Server.

The basically reverse case for request A from Fig. 1 is illustrated in Fig. 9 in the form of a flow chart and in Fig. 14 in the form of a block diagram for the arrangement that is needed to implement the process.

In this case, request A is formed 901 by the user unit XS.

Request A is returned to the transport protocol layer TP and it is there embedded 902 into the particular transport protocol format that is used. An encoded request CA, resulting from that, is now - either in user unit XS itself or in a separately provided second secure computer unit SC2 that is coupled with user unit XS via a secure channel - "unpacked" in transport protocol layer TP, in other words, it is decoded 903, which forms a decoded request DA.

In security layer SL, the decoded request DA that is supplied to it will now be subjected 904 to the provided cryptographic processes that will be described below. This results in a cryptographically processed request VA.

The cryptographically processed request VA again is supplied 905 to the transport protocol unit TP and it is encoded there, as a result of which, there is formed an encoded cryptographically processed request CVA. The encoded cryptographically processed request CVA is transmitted in a last step 906 to the program P, in other words, to the Client XC.

A development of the process according to Claim 1 is shown in Fig. 7 in the form of a flow chart and the arrangement needed to implement this process is shown in Fig. 16.

After the performance of the process steps shown in Fig. 6 to the lastly formed encoded cryptographically processed communication CVB that is transmitted to the user unit XS, the

encoded cryptographically processed communication CVB is received 701 by the at least one user unit XS or by the second secure computer unit SC2.

Using the transport protocol employed for encoding, the encoded cryptographically processed communication CVB is "unpacked," in other words, it is decoded 702 in the transport protocol layer TP of the user unit or of the second security computer unit SC2.

That forms a decoded cryptographically processed communication DVB that is now supplied to the security layer SL, which is also provided on the side of the user unit XS or the second security computer unit SC2. In the security layer SL, the decoded cryptographically processed communication DVB is subjected 703 to the particular inverse cryptographic processes. In this context, inverse means inversely with respect to the cryptographic processes that were applied in the security layer of the Client XC or of the first security computer unit SC1 upon the decoded communication DB.

The result of this cryptographic processing is an inversely cryptographically processed communication DEB, which now again is supplied to the transport protocol layer TB, where it is also again encoded 704.

The resultant encoded inversely cryptographically processed communication CEB is again supplied to the transport protocol layer TB and is decoded 705 there.

The resultant communication is now supplied to the actual Server XS, in other words, to the user unit XS, and is further processed there. A variant of the process is of course also possible in terms of directly further processing the inversely cryptographically processed communication DEB.

The basically identical development of the process according to Claim 4, that is, the same as in the previously described development for the process according to Claim 1, is illustrated in Fig. 10 along with the arrangement needed to implement the process according to Claim 5 shown in Fig. 17.

In this development again, one starts with the idea that the process steps, described in Fig. 9 up to the encoding of the cryptographically processed request VA and their transmission to the program P, have been carried out.

The transmitted encoded cryptographically processed requirement CVA is received 1001 by the program P or by the first security computer unit SC1.

In a further step 1002 using the transport protocol, one again "unpacks" the encoded cryptographically processed request CVA, in other words, it is decoded in the transport protocol layer TP. /4

Furthermore, the resultant decoded cryptographically processed request DVA is subjected 1003 to the cryptographic processing that is inverse with respect to the employed cryptographic process in the security layer SL to which it was supplied.

The resultant inversely cryptographically processed request DEA is now again coded 1004 in the transport protocol layer TP.

Then it is again decoded 1005 in the transport protocol layer TP and is supplied to program P. That is where the actual request A is further processed.

Again it is just as well possible directly to supply the decoded inversely cryptographically processed request DEA to the program P and to process it further there.

Fig. 11 describes another process that is similarly based on the common inventive idea of the processes described above.

This time, however, it is presumed that it is possible directly to insert a security layer SL between Client XC and transport protocol layer TP. This now no longer creates the need on the side of Client XC twice "to run through" the transport protocol layer TP.

This is illustrated in the arrangement in Fig. 17.

Here again, program P formulates 1101 the communication B. But communication B, however, this time is directly subjected VB, 1102 to a cryptographic process in security layer S- [sic].

The resultant cryptographically processed communication VB is supplied to transport protocol layer TP where it is encoded 1103.

The encoded cryptographically processed communication CVB is transferred 1104 to the user unit XS, there it is received 1105 by user unit XS or by the second security computer unit SC2, it is decoded in the transport protocol layer TP that is provided there into the decoded cryptographically processed communication DVB 1106.

The latter is supplied to security layer SL and is there subjected 1107 to the inverse cryptographic process or processes.

In the last two steps, the inversely cryptographically processed communication DEB is again encoded 1108 in transport protocol layer TP and is decoded 1109 in a last step.

The resultant communication B is supplied to Server XS and is further processed.

Security layer SL is illustrated in Fig. 4 for the case where it is possible to insert the security layer SL between the transport layer TP and the library routines XL.

This time, however, looking at the special example which in no way whatsoever restricts the general validity, unsecured read, write, readv, writev connect and accept communications are "secured" by the cryptographic process provided in security

layer SL. This is done by applying the provided cryptographic processes upon the particular communication B or request A. The communications that are "secured" by security layer SL are marked in Fig. 4 by an asterisk *.

The described cryptographic securing of the communication of an application with a window system via a network, on the one hand, presupposes the exchange of the cryptographic codes and, on the other hand, is based on a reciprocal authentication of the two communications partners.

For this authentication, one can advantageously employ asymmetrical cryptographic processes together with certificates that contain public codes. By suitable definition of the identity characteristics in the certificate, it is possible to identify and authenticate services such as applications or window service programs beyond the mere computer address in the network. Such identity characteristics that go beyond the network address for the differentiation of various application programs of a computer, for example, can be the name of the service owner on a multiuser system.

The reciprocal authentication and the code exchange are carried out in an initialization phase to build up the secure connection.

As a further development of the invention-based process, it is advantageous on the side of the window service program, in

other words, the user unit XC, to carry out an access check on the basis of the authenticated identity of program P. The authenticated identification information can go beyond the computer address of program P; therefore, an access check can differentiate between various programs P of a computer and can thus control the buildup of the connection.

An advantageous application of the described security procedure can be found in the exchange of application data between a program P and a window service program, in other words, a user unit XC, where only one network connection that is not worthy of confidence can be switched between both of them.

This scenario is especially important to the above-described CSCW systems that do the Application Sharing. Here the participating window service programs of the user units XC are often in different company networks and can exchange data with the application or the Application Sharing component only via public networks.

Considerable security problems are connected with the operation of the known window system and they are described in [6], [7]. Due to the considerable risk potential that is tied to the window system known from [1], the operators of company networks as a rule do not allow such window service programs to cooperate with applications outside the company network. This is intended to predict internal company information and data

inventories. This protection is provided by the so-called Firewalls at the network transition between the in-house network and the outside networks. By filtering data packets on the transport system level, those outside networks prevent external application programs from accessing the in-house window service programs.

These customary precautions, however, prevent the use of synchronous CSCW systems that are based on the following: Users at various sites and in various companies together cooperate via a synchronous CWCW system and together work with application programs.

On the basis of the described security process for application data, one can construct a program for a Firewall, which makes it possible to allow in-house window service programs securely to communicate with outside application programs:

This special program is based, on the one hand, on the described security expansion to protect application data in window systems and, on the other hand, on a gating component for application data. The gating component can be derived directly from the Application Sharing Component ASC because in this case there is no request for multiplexing and demultiplexing. /5

These two components (security service program and gating component) form a special Firewall security service program by

means of which it becomes possible from an external application program to request a specific authentication as well as to subject it to an access check before the gating component establishes the connection to the in-house window service program and then switches the connection through. The subsequent data exchange between the external application program and the Firewall security service program is protected by cryptographic mechanisms.

By operating packet filters in the Firewall, one can force external application programs first of all to establish connection with the described security service program.

The corresponding process, considering the "switch of roles" between program and user unit XS, in other words, for request A, is illustrated in Fig. 12 along with the arrangement needed for its implementation shown in Fig. 18.

Here one naturally assumes that security layer SL can be inserted on the side of the user unit XS between the user unit XS and the transport protocol layer TP.

On the basis of this assumption, user unit XS thus forms 1201 the request A. This request is directly subjected 1201 to the cryptographic process VA in security layer SL.

The cryptographically processed request VA is encoded 1203 in the transport protocol layer TP and subsequently thereto the

encoded cryptographically processed request CVA is transmitted 1204 to the program P.

There it is received 1205 by program P or by the first security computer unit SC1. In a transport protocol layer TP, which is also provided there, it is now decoded 1206 to the decoded cryptographically processed request DVA.

In security layer SL to which it is supplied in an additional step, the decoded cryptographic request is subjected 1207 to the inverse cryptographic process. The resultant inversely cryptographically processed request DEA is again "packed up" in the transport protocol layer TP, in other words, it is encoded 1208.

The encoded inversely cryptographically processed request CEA is again decoded 1209 in an additional step in transport protocol layer TP and the resultant request A that is now cryptographically "secured" is supplied to the program and is further used by the program P.

Various possibilities for implementing the cryptographic processes to be used in security layer SL are illustrated in Fig. 8.

First of all, it is possible to apply encoding processes 81 in security layer SL. In that way, one can achieve a confidentiality or integrity of the exchanged communications B or requests A.

It is furthermore provided that authentication mechanisms 82 can also be used in security layer SL. These mechanisms make it possible to verify identity data of the communications partners in the network. These authentication mechanisms have a special meaning in the context, for example, of the Transport Control Protocols (TCP) or also the User Datagramm Protocols (UDP) because they do not display any authentication mechanisms for senders and recipients.

The implementation of access check mechanisms 83 that are based on the authentication processes also offers additional protection for the access to the window service program in a Client-Server window system.

The processes described above naturally can also very advantageously be applied to multiuser systems.

The way in which the Client-Server window system described in [1] can be expanded into a multiuser system is described, for example, in [3], [4], [5].

The resultant situation with an additional multiplexer component ASC and several user units XSi, where an index I definitely identifies each user unit XSi and is a natural number in the range from 1 to n, is illustrated in Fig. 2.

Here, the requests Ai are in the known manner combined by the individual user units XSi and communication B is transmitted to the individual user units XSi as copies of communication Bi.

The invention-based process in this context naturally are performed for each individual connection between client XC and each user unit XSi.

This "multiuser environment" is described in even greater detail in Fig. 3. In this practical implementation, the requests Ai correspond to so-called Xrequests and the communications Bi correspond to the so-called Xreplies, Xevents, Xerrors. The application ANW accesses the system resources SR via the system calls SC.

By way of a further development of the process, it is advantageous at the start of the process to provide for an initialization phase in which, for example, one performs a code exchange as well as a bilateral authentication between a user unit XS of the user units XSi and the program P.

The most varied processes for code exchange are known in this case to the expert. As an example of an initialization phase that can be employed in the invention-based process, we propose the following procedure:

The process for code exchange, described below, is generally carried out between client XC and a user unit XS. The multiplexer component ASC in this context is to be considered as a special component of client XC.

Assuming that the multiplexer component ASC has an application certificate and that the user units, in other words,

the servers XSi in each case do have a user certificate which in each case unambiguously are associated with the user units, the multiplexer component ASC then generates a first random number.

After a transport connection between the multiplexer component ASC and the particular server XSi has been filled up, the multiplexer component ASC transmits a first negotiation communication to the user unit, which at least displays the following components: /6

- the program certificate,
- the first random number,
- a first proposal for a cryptographic process that is to be used further on and
- a digital signature that is formed at least via the first random number as well as the first proposal.

The first negotiation communication is received by the particular user unit, in other words, the server XSi.

The user unit XSi checks the program certificate for correctness.

The digital signature is also checked out.

If the check on the program certificate and the digital signature produces a positive result, then the user unit XSi further checks whether the proposed cryptographic algorithms that were proposed in the first negotiation communication can continue to be used to secure the transmission.

When the user unit XSi cannot support the proposed cryptographic algorithms, then the user unit, in other words, the server XSi, formulates a second proposal in a second proposal communication and transmits it to the multiplexer component ASC. The second proposal displays cryptographic processes that are supported by the user unit XSi. These [processes] are now proposed to the multiplexer component ASC as cryptographic processes to be employed as the procedure is further pursued for this logical connection between the multiplexer component and the user unit XSi.

The second proposal communication has at least the following components:

- the user certificate of the particular server XSi,
- a second random number that was generated by the user unit XSi itself,
- the second proposal,
- a digital signature that in each case would be formed at least via the first random number, the second random number as well as the second proposal.

The second proposal communication is transmitted to the multiplexer component ASC.

If the cryptographic algorithms given in the first proposal are supported by the user unit XSi, then the user unit XSi

formulates a confirmation communication and sends it to the multiplexer component ASC.

The confirmation communication has at least the following components:

- the user certificate,
- the second random number,
- a positive confirmation and
- a digital signature formed in each case at least via the first random number, the second random number and the positive confirmation.

The confirmation communication is sent to the multiplexer component ASC.

The multiplexer component ASC receives the negotiation communication or the confirmation communication and the multiplexer component ASC checks to see whether the user certificate as well as the digital signature are correct.

Furthermore, if the check yields a positive result and if the received communication was the confirmation communication, the multiplexer component ASC will generate a first session code, considering the agreed-upon cryptographic algorithms for a subsequent useful data transmission phase.

A first session code communication is formed from the first session code and is sent to the user unit XSi, which at least has the following components:

- the first session code that is encoded with a public code of the server XSi,

- a specification of the cryptographic processes to be employed,

- a digital signature that is formed at least via the first random number, the second random number, the first session code as well as the specification of the cryptographic processes to be employed.

If the multiplexer component ASC received the second negotiation communication and if the check on the user certificate and the digital signature or the hash value of the second negotiation communication yielded a positive result, then the multiplexer component ASC checks to see whether the cryptographic algorithms proposed in the second negotiation communication are supported by the multiplexer component ASC for the implementation of additional cryptographic processes.

When the proposed cryptographic processes are supported by the multiplexer component ASC, then a first session key is generated, considering the agreed-upon cryptographic algorithms for the following useful data transmission phase.

Furthermore, as described above, a first session code communication is sent to the multiplexer component ASC using the first session code.

This above-described procedure for "negotiating" the cryptographic processes that are to be employed is repeated until such time as both the user unit XSi and the multiplexer component ASC accept the last-proposed cryptographic processes.

In user unit XSi, the first session code is determined, employing a private code of the user unit XSi. Furthermore, the digital signature of the first session code communication is checked out.

Besides, if the check of the digital signature supplies a positive result, a second session code communication is formulated, employing a second session code that is formulated by the user unit XSi.

The second session code communication displays at least the following components:

- the second session code encoded with a public program code of the multiplexer component ASC and

- a digital signature that is formed at least via the first random number, the second random number, the second session code or a hash value formed via the same components. /7

The multiplexer component ASC receives the second session code communication and determines the second session code. The digital signature or the hash value of the second session code communication is now checked out.

If the check on the digital signature yielded a positive result, then the exchange session keys are employed in the following useful data transmission phase for the purpose of encoding the useful data. Here, each participating instance employs the session key that it generated itself for transmitting useful data, while the received session key is employed exclusively to receive useful data.

Other cryptographic processes for code exchange or to form the session code for useful data encoding can be employed in the context of the invention-based process without any restrictions.

The invention-based processes can be employed very advantageously in the following scenario.

Highly confidential information are exchanged between cross-linked computers in many private networks. Here the private network itself is mostly very well secured against the outside world, for example, by so-called Firewalls [6].

If a computer connected to the secured private network would like to communicate with a computer that can be reached outside that network only via an unsecured channel, for example, a computer that can be reached only via the Internet IN, then, so far, there was a big problem: No secure communication is possible in the Client-Server window systems based on [1].

In particular, one encounters the following problem: Other applications can be attacked via the window service program. To

prevent snooping on in-house information, company networks as a rule are not allowed to operate a window service program outside the company network. This generally customary restrictions, in particular, hinders synchronous CSCW systems that are based on Application Sharing.

These problems are described in detail, for example, in [6], [7].

These problems need not necessarily involve a communication that overlaps a local network; instead, this, for example, can also involve a secure Corporate Network CN where a communication partner wants to communicate with another communication partner of another company via the computer, for example, in a CSCW system.

By means of the invention-based processes, it is now possible when these processes are employed in a Firewall SC1, SC2 of the local network or of the Corporate Networks CN [sic; verb missing in original], whereby precisely the Firewall in each case is to be considered as a first securing computer unit SC1 or as a second securing computer unit SC2 (see Fig. 3).

The following publications were cited in this document:

[1] R. Scheifler et al., "The X Window System," ACM Transactions on Graphics, Vol. 5, № 2, pp. 79 to 109, April 1986.

[2] S. Garfinkel et al., "Practical UNIX Security," O'Reilly & Associates, Inc., ISBN 0-937175-72-2, pp. 221-253, 1991.

[3] H. Abdel-Wahab et al., "Issues, Problems and Solutions in Sharing X Clients on Multiple Displays," Internetworking: Research and Experience, Vol. 5, pp. 1 to 15, 1994.

[6] [sic; [4]] D. Garfinkel at al., "HP Shared X: A Tool for Real-Time Collaboration," Hewlett-Packard Journal, pp. 23 to 36, April 1994.

[5] J. Baldeschwieler et al., "A Survey of X Protocol Multiplexors," Swiss Federal Institute of Technology, Computer Engineering and Networks Laboratory (TIK), ETH-Zentrum, Zurich, 1993.

[6] S. Bellovin et al., "Network Firewalls," IEEE Communications Magazine, pp. 60 to 57 [sic], September 1994.

[7] G. Treese et al., "X Through the Firewall and Other Application Relays," Summer Usenix, 1993, 21 to 25, June, Cincinnati, pp. 87 to 98, 1993.

Claims

1. Process for the cryptographic securing of computer-assisted digital communication between a program (P) and at least one user unit (XSi),

- where program (P) forms (601) a communication (B),

- where a computer unit on which the program (P) is processed or a first securing computer unit (SC1) encodes (602) the communication (B) with a transport protocol (CB),

- where the encoded communication (CB) is decoded (603) using the transport protocol (DB),

- where the decoded communication (DB) is subjected (604) to a cryptographic process (VB),

- where the cryptographically processed communication (VB) is encoded (605) with the transport protocol (CVB) and

- where the encoded cryptographically processed communication (CVB) is transmitted (606) to the at least one user unit (XSi).

2. Process according to Claim 1,

- where the encoded cryptographically processed communication (CVB) is received (701) by the at least one user unit (XSi) or by a second securing computer unit (SC2),

- where using the transport protocol the encoded cryptographically processed communication (CVB) is decoded (702) (DVB),

- where the decoded cryptographically processed communication (DVB) is subjected (703) to a cryptographic processing (DEB) that is inverse with respect to the cryptographic process,

- where the inversely cryptographically processed communication (DEB) is encoded (704) with the transport protocol (CEB), and

- where the encoded inversely cryptographically processed communication (CEB) is decoded (705) using the transport protocol.

3. Process according to Claim 1, /8

- where the encoded cryptographically processed communication (CVB) is received by the at least one user unit (XSi),

- where, using the transport protocol, the encoded cryptographically processed communication (CVB) is decoded (DVB) and

- where the decoded cryptographically processed communication (DVB) is subjected to a cryptographic processing (DEB) that is inverse with respect to the cryptographic process.

4. Process for the cryptographic securing of computer-assisted digital communication between a program (P) and at least one user unit (XSi),

- where a user unit (XSi) formulates (901) a request (A),

- where the user unit (XSi) or a second securing computer unit (SC2) encodes (902) the request (A) with a transport protocol,

- where the encoded request (CA) is decoded (903) using the transport protocol (DA),

- where the decoded request (DA) is subjected (904) to a cryptographic process (VA),

- where the cryptographically processed request (VA) is encoded (905) with the transport protocol (CVA) and

- where the encoded cryptographically processed request (CVA) is transferred (906) to the program (P).

5. Process according to Claim 4,

- where the encoded cryptographically processed request (CVA) is received (1001) by the program (P) or by a first securing computer unit (SC1),

- where, using the transport protocol, the encoded cryptographically processed request (CVA) is decoded (1002) (DVA),

- where the decoded cryptographically processed request (DVA) is subjected (1003) to a cryptographic processing (DEA) that is inverse with respect to the cryptographic process -

where the inversely cryptographically processed request (DEA) is encoded (1004) with the transport protocol (CEA) and

- where the encoded inversely cryptographically processed request (CEA) is decoded (1005) using the transport protocol.

6. Process according to Claim 4,

- where the encoded cryptographically processed request (CVA) is received by the program (P),

- where, using the transport protocol, the encoded cryptographically processed request (CVA) is decoded (DVA) and

- where the decoded cryptographically processed request (DVA) is subjected to a cryptographic processing (DEA) that is inverse with respect to the cryptographic process.

7. Process for cryptographic securing of computer-assisted digital communication between a program (P) and at least one user unit (XSi),

- where program (P) formulates (1101) a communication (B),

- where the communication (B) is subjected (1102) to a cryptographic process (VB),

- where the cryptographically processed communication (VB) is encoded (1103) with the transport protocol (CVB),

- where the encoded cryptographically processed communication (CVB) is transmitted (1004) to at least one user unit (XSi),

- where the encoded cryptographically processed communication (CVB) is received (1105) by at least one user unit (XSi) or of a second securing computer unit (SC2),

- where, using the transport protocol, the encoded cryptographically processed communication (CVB) is decoded (1106) (DVB),

- where the decoded cryptographically processed communication (DVB) is subjected (1107) to a cryptographic processing (DEB) that is inverse with respect to the cryptographic process,

- where the inversely cryptographically processed communication (DEB) is encoded (1108) using the transport protocol (CEB) and

- where the encoded inversely cryptographically processed communication (CEB) is decoded (1109) using the transport protocol.

8. Process for the cryptographic securing of computer-assisted digital communication between a program (P) and at least one user unit (XSi),

- where at least one user unit (XSi) formulates (1201) a request (A),

- where the decoded request (DA0 is subjected (1202) to a cryptographic process (VA),

- where the user unit (XSi) or a second securing computer unit (SC2) encodes (1203) the cryptographically processed request (A) with a transport protocol (CA),

- where the encoded cryptographically processed request (CVA) is transferred (1204) to the program (P),

- where the encoded cryptographically processed request (CVA) is received (1205) by the program (P) or by a first securing computer unit (SC1),

- where, using the transport protocol, the encoded cryptographically processed request (CVA) is decoded (1206) (DVA),

- where the decoded cryptographically processed request (DVA) is subjected (1207) to a cryptographic processing (DEA) that is inverse with respect to the cryptographic process,

- where the inversely cryptographically processed request (DEA) is encoded (1208) with the transport protocol (CEA) and /9

- where the encoded inversely cryptographically processed request (CEA) is decoded (1209) using the transport protocol.

9. Process according to one of Claims 1 to 8,

- where the cryptographic processing is performed (81) at least by one encoding of the request (Ai) and

- where the inverse cryptographic processing is carried out at least by one decoding of the request (Ai).

10. Process according to one of Claims 1 to 9,

- where the cryptographic processing is carried out (82) at least by means of authentication mechanisms for the request (Ai) and

- where the inverse cryptographic processing is carried out at least by inverse authentication mechanisms for the request (Ai).

11. Process according to one of Claims 1 to 10,

- where the cryptographic processing is carried out (83) at least by access control mechanisms for request (Ai) and

- where the inverse cryptographic processing is carried out at least by inverse access control mechanisms for request (Ai).

12. Process according to one of Claims 1 to 11, where at the start of the process, a cryptographic initialization phase with formation of a session code is performed for each connection of a user unit (XSi) with the program (P).

13 pages of drawings.

[Please insert Fig. 6].

[Key: 601) Communication B is formulated by program P; 602) Communication is encoded in transport protocol layer TP; 603) Communication is decoded in transport protocol layer TP; 604) Communication is subjected to cryptographic processes; 605) Communication is encoded in transport protocol layer TP; 606) Encoded communication is transferred to user unit XS].

[Please insert Fig. 8].

[Key: 80) Cryptographic process in security layer SL; 81) Encoding; 82) Authentication mechanisms; 83) Access control mechanisms].

[Please insert Fig. 7].

[Key: 601) Communication B is formulated by program P; 602) Communication is encoded in transport protocol layer TP; 603) Communication is decoded in transport protocol layer TP; 604) Communication is subjected to cryptographic processes; 605) Communication is encoded in transport protocol layer TP; 606) Encoded communication is transferred to user unit XS; 701) Encoded communication is received by user unit XS; 702) Received communication is decoded in transport layer TP; 703) Decoded communication is processed in an inversely cryptographic manner in security layer SL; 704) Inversely processed communication is encoded in transport layer TP; 705) Encoded communication is decoded in transport layer TP].

[Please insert Fig. 9].

[Key: 901) Request A is formulated by user unit XS; 902) Request A is encoded in transport protocol layer TP; 903) Request A is decoded in transport protocol layer TP; 904) Request A is subjected to cryptographic processes; 905) Request A is encoded in transport protocol layer TP; 906) Encoded request is transferred to program P].

[Please insert Fig. 10].

[Key: 901) Request A is formulated by user unit XS; 902) Request A is encoded in transport protocol layer TP; 903) Request A is decoded in transport protocol layer TP; 904) Request A is

subjected to cryptographic processes; 905) Request A is encoded in transport protocol layer TP; 906) Encoded request A is transferred to program P; 1001) Encoded request A is received by program P; 1002) Received request A is decoded in transport layer TP; 1003) Decoded request A is processed in an inversely cryptographic manner in security layer SL; 1004) Inversely processed request A is encoded in transport layer TP; 1005) Encoded request A is decoded in transport layer TP].

[Please insert Fig. 11].

[Key: 1101) Communication B is formulated by program P; 1102) Communication B is subjected to cryptographic processed; 1103) Communication B is encoded in transport protocol layer TP; 1104) Encoded communication B is transferred to user unit XS; 1105) Encoded communication B is received by user unit XS; 1106) Received communication B is decoded in transport layer TP; 1107) Decoded communication B is processed in an inversely cryptographic manner in security layer SL; 1108) Communication B is encoded in transport protocol layer TP; 1109) Communication B is decoded in transport protocol layer TP].

[Please insert Fig. 12].

[Key: 1201) Request A is formulated by user unit XS; 1202) Request A is subjected to cryptographic processed; 1203) Request A is encoded in transport protocol layer TP; 1204) Encoded request A is transferred to program P; 1205) Encoded request A

is received by program P; 1206) Received request A is decoded in transport layer TP; 1207) Decoded request A is processed in an inversely cryptographic manner in security layer SL; 1208) Request A is encoded in transport protocol layer TP; 1209) Request A is decoded in transport protocol layer TP].